

ASp

Ian Greenway

COLLABORATORS

	<i>TITLE :</i> ASp		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Ian Greenway	January 13, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ASp	1
1.1	ASp Guide	1
1.2	Changes since V0.82b	1
1.3	Changes since V0.82	2
1.4	Changes since V0.81	2
1.5	Introduction	3
1.6	What's a Spectrum?	3
1.7	ASp Feature Guide	4
1.8	ASp System Requirements	4
1.9	ASp Installation	5
1.10	Which ROM files to use?	5
1.11	ASp Quick Start Guide	6
1.12	What features are coming up?	7
1.13	Detailed Usage Guide for ASp.	7
1.14	Basic Operation Overview	8
1.15	How to Quit ASp	9
1.16	Keyboard layout information	9
1.17	Types of Joystick available in ASp	10
1.18	Kempston Joystick operation	10
1.19	Smart Kempston Joystick operation	10
1.20	Cursor Joystick operation	11
1.21	Sinclair 1 Joystick operation	11
1.22	Sinclair 2 Joystick operation	11
1.23	Custom Keys Joystick operation	11
1.24	Hardware Direct port access for Joysticks	12
1.25	Auto-detect lowlevel.library port access	12
1.26	Lowlevel.library port access for CD32 gamepad	12
1.27	Lowlevel.library port access for joystick	12
1.28	Gameport.device port access for joystick	12
1.29	Emulation Speed adjustment	12

1.30	Loading Snapshots	13
1.31	Saving Snapshots	13
1.32	Using tape files	14
1.33	Project->Grab Display...	14
1.34	Description of .Z80 files	14
1.35	Description of .SNA files	14
1.36	Description of .SP files	15
1.37	Description of .TAP files	15
1.38	Description of .TZX files	15
1.39	TZX Blocks supported by ASp	15
1.40	Emulated Machine Type Selection	16
1.41	Audio Configuration	17
1.42	Settings->Audio->Select AHI Mode	17
1.43	Audio Channel Mode	17
1.44	Audio Buffer Size	18
1.45	Settings->Audio->AY Volume steps	18
1.46	Settings->Audio->Quiet 48K Beeps	18
1.47	Settings->Audio->Swap Stereo Channels	18
1.48	Audio Save option	18
1.49	Video configuration options	19
1.50	Emulation Display type selection	19
1.51	More information about palette selection.	20
1.52	Screen Mode selection	20
1.53	Screen Centring options	20
1.54	Colour Zero Border option	21
1.55	Force Chip-mem Screen option	21
1.56	Blank Mouse Pointer option	22
1.57	Remember display window position	22
1.58	Window mode snap to grid.	22
1.59	Why align the window to 32 pixels?	22
1.60	Automatic Calibration of rendering routines.	23
1.61	Frame Skip function	23
1.62	Miscellaneous configuration options	23
1.63	Settings->Misc->Add AY chip to 48K model	24
1.64	Settings->Misc->Keep Kempston Interface	24
1.65	Settings->Misc->Flash-Load Standard Blocks	24
1.66	Settings->Misc->Delay during Flash-Load	24
1.67	Settings->Misc->Enable F-keys	24
1.68	Settings->Misc->Load snapshot prefs	25

1.69	Settings->Misc->Grab display with border?	25
1.70	Settings->Misc->Compress IFF grabs?	25
1.71	Loading and Saving User Prefs Settings	25
1.72	ASp initialisation behaviour	25
1.73	ASp ToolTypes	26
1.74	Specification of type LITERAL tooltypes	27
1.75	Specification of type BOOLEAN	27
1.76	Specification of type STRING tooltypes	27
1.77	Specification of type NUMERIC tooltypes	28
1.78	ASp Command-Line arguments	28
1.79	AREXEXPORTNAME	28
1.80	EXTIMAGEVIEWER	28
1.81	EXTTEXTVIEWER	28
1.82	LOADPATH Argument	29
1.83	SAVEPATH Argument	29
1.84	ROMPATH Argument	29
1.85	ROMPAGE0 Argument	29
1.86	ROMPAGE1 Argument	29
1.87	ROMPAGE2 Argument	30
1.88	ROMPAGE3 Argument	30
1.89	ROMPAGE4 Argument	30
1.90	48KISSUE2 Argument	30
1.91	ADAPTIVEFS Argument	30
1.92	ADD48KAY Argument	31
1.93	AUDBUFFER Argument	31
1.94	AUDIOMODE Argument	31
1.95	AYCHAR Argument	31
1.96	BLANKPOINTER Argument	31
1.97	CENTREX Argument	32
1.98	CENTREY Argument	32
1.99	COL0BORD Argument	32
1.100	DISPLAY Argument	32
1.101	FLASHLOAD Argument	32
1.102	FRAMESKIP Argument	32
1.103	JOYPORTACCESS Argument	33
1.104	JOYSTICK Argument	33
1.105	KEEPKEMPSTON Argument	33
1.106	MACHINE Argument	33
1.107	QUIET48 Argument	34

1.108SCREENDEPTH Argument	34
1.109SCREENMODE Argument	34
1.110SPEED Argument	34
1.111SWAPSTEREO Argument	34
1.112TAPDELAY Argument	35
1.113USEFKEYS Argument	35
1.114The ASp Tape file browser	35
1.115ASp ROM Configuration GUI	35
1.116Copyrights on Spectrum Software	36
1.117ARexx port functions	36
1.118Running an ARexx script from within ASp	36
1.119ASp ARexx Commands	37
1.120ASp ARexx important information	37
1.121ARexx command: HELP	38
1.122ARexx command: GETVERSION	38
1.123ARexx command: RUN	39
1.124ARexx command: STOP	39
1.125ARexx command: RESET	39
1.126ARexx command: QUIT	39
1.127ARexx command: LOAD	40
1.128ARexx command: SAVE	40
1.129ARexx command: RELOAD	40
1.130ARexx command: INSERT	41
1.131ARexx command: REWIND	41
1.132ARexx command: POKE	41
1.133ARexx command: PEEK	41
1.134ARexx command: READRAM	42
1.135ARexx command: WRITERAM	42
1.136ARexx command: SETPAGE	42
1.137ARexx command: GETPAGE	43
1.138Demo script: LoadSCR.asprx	43
1.139Demo script: SaveSCR.asprx	43
1.140Demo script: POKEr.asprx	43
1.141Copyrights and Distribution	43
1.142Credits	44
1.143Version History	45
1.144Frequently Asked Questions	50
1.145Contended RAM	54
1.146Information about ASp's use of mmu.library	55

1.147	What's an MMU?	55
1.148	About the XFD package	55
1.149	About the AHI package	56
1.150	World Of Spectrum	56
1.151	ASp Support Web Pages	56
1.152	Contacting the Author	56
1.153	Index	57

Chapter 1

ASp

1.1 ASp Guide

Amiga Spectrum emulator

V0.84

(c) Ian Greenway 1998-2004

[Summary of changes since V0.82b](#)

[Introduction](#)

[Features](#)

[Requirements](#)

[Installation](#)

[Quick start](#)

[Full usage](#)

[Credits](#)

[Contact](#)

[Version History](#)

[To Do](#)

[FAQ's](#)

[Copyright and distribution](#)

[Index](#)

1.2 Changes since V0.82b

Will now run in 48K mode if mmu.library not available.

Now uses AHI for sound. 16-bit resolution.

AY chip amplitude and frequency resolution enhanced.

Complete rewrite of graphics update code. Static screens should be less load. As a side-effect, the "higher accuracy timing" mode of previous versions is always active.

Changed parts of the emulation core for speed.

(Combination of the previous two offer up to 20% speedup with 68060.)

Fixed problem with ROM Manager not releasing slots.

TZX loader resolution increased. Fixes some loaders at expense of RAM.

Changed prefs file format to tootype-alike. Human readable and will allow backwards compatibility.

The DAA instruction did not behave correctly with some illegal input values. Now verified to behave identically to a real Z80.

Added ability to grab the display to IFF.

CTRL key now acts as extend-shift.

Miscellaneous bugfixes and code tidying.

[Changes since V0.82](#)

[Full version history](#)

1.3 Changes since V0.82

Fixed port \$7FFD address decode problem.

Changed some emulation routines for speed. Thanks to Neil Whitehead for tests.

Revert to original use of [lowlevel.library](#) , as V0.82 seems excessively slow.

[Issue 2](#) hardware may not have worked in V0.82!

Fix a (harmless) buffer overrun in init.

[Changes since V0.81](#)

[Full version history](#)

1.4 Changes since V0.81

Added a [ROM Configuration GUI](#).

Added [.SP](#) load and save ability. (48K-only format.)

Added support for more [TZX blocks](#) :

Archive Info, User Select, Instructions, Picture.

Use of [lowlevel.library](#) now optional: only required for CD32 pad. User can now specify [lowlevel.library access mode](#) or use direct hardware access. Will hopefully avoid various incompatibility problems previously seen.

Uses new mode of [mmu.library](#) which should be compatible to FScreen. (Requires V43 of [mmu.library](#).)

Now allows drag-n-drop of TAP/TZX files.

Pressing keys in the [tape-browser](#) now navigates the tape, not activates the ZX keyboard.

Added a simple status bar for user feedback.

Now also allows damaged TAP files to be loaded, with confirmation.

Total rewrite of [Flash-loader](#) for Standard Blocks. Fixes some incompatibilites. (eg. Lemmings.)

No longer holds locks on loaded files.

Add 48K Issue2 support.

Will now decrunch ROM files as well as snapshots and tapes.

Changed the format selection system for [saves](#) slightly.

Made changes to Z80 emulation of 8-bit ADD, SUB, INC, DEC and CP which may offer some speedup on '030. Penalty is use of 128Kb more RAM...

Fix bug introduced in V0.81 which did not respect the lock/disable bit of port \$7FFD for 128K page-mapping. :-/

Assorted internal reworking of string and file handling.

Misc bugfixes.

For earlier information, see [full version history](#)

1.5 Introduction

ASp is a [Sinclair Spectrum](#) emulator for Amiga computers. It models the behaviour, appearance and sound of the Sinclair computer on your Amiga. Spectrum software can be loaded and saved in a variety of popular formats.

ASp provides a comprehensive and accurate emulation of the 48K, 128K and 128K Plus 2 Spectrum models with clean, configurable stereo sound and authentic graphics. It is OS-friendly, and can run on systems with graphics and/or sound cards, as well as OCS/AGA. A reasonably powerful 68K processor is required for useful performance.

There are many Spectrum emulators available for many computer platforms. Some favour speed, accuracy or features. The design philosophy for ASp is to create an easy-to-use emulator which runs almost all Spectrum software and has a good selection of features. It is one of the few emulators available for the Amiga at this time that can emulate the 128K models.

In addition, the latest release has been specifically modified to offer better compatibility to newer Amiga systems, such as an "AmigaOne" running OS4, as well as potentially other systems like MorphOS or UAE.

(Note, for this release, OS4/MOS/UAE operation is limited to the 48K model.)

The primary features list is [here](#).

1.6 What's a Spectrum?

What is a Sinclair ZX Spectrum?

No... I have difficulty in believing you don't know, surely... What on earth did you download this for?

Spectrums were 8-bit computers devised by Sir Clive Sinclair and were very popular during the 1980's. It has been said that the 48K Spectrum is the biggest selling single model of home computer ever. That includes PC's, since no two models are the same, and they're not "home" computers! :-)

So much for the hype... What do they do?

Spectrums had a modest specification for the time, being mainly designed to be cheap and accessible. The CPU was a 3.5MHz Z80 (pretty much the best 8-bit CPU ever made), with just 16Kb of RAM in the early models, and 48Kb in the later, more popular model, a 16Kb ROM and a simple tape interface. The graphics were, technically, 16-colour, but limited to 2 colours per character block, with bright and darker shades of a basic 8 colours making up the 16. Sound was limited to a small loudspeaker being toggled from a control line. As with many machines of the day, a slow but comprehensive BASIC interpreter was available at power-on.

There were many quirky aspects of the machine that provided character to the Spectrum, such as the initially off-putting keyword entry system, the much-discussed graphic "attribute clash", the infamous rubber keyboards of the 48K model, and the way the case got so hot you could barely touch it. Yet possibly the most endearing aspect of all was the way some very clever people managed to squeeze the most amazing sound and graphics out of such a simple design to make extremely playable games.

Later models boasted 128Kb of memory and a 32Kb or more ROM. Higher speed tape drives and disk drives were available as accessories, and a plethora of interfaces and add-ons such as joysticks, printers, modems and other ingenious devices were invented.

The rivalry between the Spectrum and the Commodore 64 was the driving force for the explosion in the home computer gaming market during the early and mid 80's, providing a rich arena for the more powerful 16-bit machines like Amiga and Atari to enter. This next generation of computers hailed the end of the commercial viability of the 8-bit machines, although Amstrad, now the

owners of the Sinclair Spectrum technology, had made a couple of attempts to re-vamp the machines, in the form of the "Plus 2", "Plus 2A" and "Plus 3", with only modest success.

Despite even the next wave of computers, 32-bit, to arrive on the scene, there was still a massive following of nostalgia for the old Spectrums. These new machines offered the opportunity to create full-speed emulation of the old favourites and many emulators have been written. Amstrad took the unheard-of step of declaring the Spectrum technology free for emulator use: a generous move that has paid huge dividends in keeping the scene very much alive.

Games and computer technology have now moved on. The generation who fondly remembers the 8-bit days have been replaced with the new voices of those who think even 32-bit is out of date. Despite that, you can still see the roots of most modern game concepts in those deceptively clunky Speccy games, which you can still play for "five minutes" and discover an hour has unaccountably passed...

1.7 ASp Feature Guide

ASp headline feature-set:

=====

Small, easy to use, authentic emulation of 48K/128K/+2 Sinclair Spectrums.

System friendly and compliant, fully multitasking.

Font-sensitive GadTools GUI.

Can run multiple copies simultaneously.

Comprehensive, speed-regulated Z80-CPU core which includes all "undocumented" instructions and most undocumented flag behaviour.

Genuine appearance Spectrum display with full colour and support for BRIGHT, FLASH and rainbow effects.

Fully retargetable display rendering to Screen or Window on Workbench.

Clean, stable, configurable stereo sound (48K beeper and 128K AY-3-8912 chip).

Support for Kempston, Cursor, Sinclair and "key" joystick types.

Loads .SNA, .Z80, .SP, .TAP and .TZX files.

AppWindow drag-and-drop support.

Tape browser.

Configurable ROM files.

Will transparently decrunch compressed files using the XFD package.

Saves .SNA, .SP and .Z80(V2) files.

ARexx port for remote control and scripted operation.

1.8 ASp System Requirements

In order to run ASp at all, you absolutely NEED the following as a minimum:

- 68020 or better.
- Around 3Mb of free RAM.
- Workbench 3.0+
- The Spectrum 48, 128 and Plus 2 ROM files. -- **Help!**

To get full features and performance, you should have:

- For full Spectrum speed: 33MHz 68040 or better.

- For audio output: [AHI V4+](#) installed.
- For 128K features: [mmu.library V43+](#) , and a supported MMU.
- For CD32 pad support: [lowlevel.library](#) from OS3.1+
- For transparent file decompression : The [XFD package](#)
- For automatic scripts: ARexx, with REXXMAST running.

For AGA users, it is recommended to install:

- [BlazeWCP](#) by Rick Pratt : Faster screen updates when using AGA.

1.9 ASp Installation

ASp has no installer script!

Just copy the main program to wherever you want. For maximum ease of use, also copy the icon and any subdirectories.

You then need to install the ROM files, either by copying the supplied files into the same directory as the executable, or by following the information [here](#) .

You will also need to have [mmu.library](#) installed if you wish to use 128K features. See the MuLib distribution for details.

1.10 Which ROM files to use?

ASp requires ROM image files of the Spectrum 48K, 128K and 128K+2 computers.

These files are freely available from the [ASp web pages](#) , [World Of Spectrum](#) , and other public resources.

Unlike ROM images for most other systems, this IS legal! Amstrad Plc. have kindly granted permission for free distribution of the Spectrum ROM images for which they hold copyright, for use with emulation software. Amstrad Plc. retain copyright while allowing redistribution.

You can use any Spectrum ROM files with ASp, including those which have been "bugfixed" or otherwise modified. As of V0.82, ASp will decrunch the files if required.

ASp has a number of internal default patterns of ROM files that suit some popular file names, including those in the above distribution. If you wish to change which file(s) are used, there is a series of hierarchical overrides that can be specified.

ASp will look for the following files in order, using the first ones it finds:

All ROMs in one file: All the required ROM images in sequence, no other files are used.

-> The 80Kb file from the "ROMPAGE0" argument, if supplied.

OR: Any combination of the following...

The Spectrum 48K ROM:

If supplied: The 16Kb file from the "ROMPAGE0" argument.

Else: "Spectrum.rom" (16Kb) (1st internal default)

Or: "48.ROM" (16Kb) (2nd internal default)

The Spectrum 128K ROM(s):

Either a single 32Kb file or TWO 16Kb files, as follows:

If supplied: The 32Kb file from the "ROMPAGE1" argument.

Else: "Spec128.rom" (32Kb) (1st internal default)

Alternatively:

If supplied: The 16Kb file from the "ROMPAGE1" argument.

Else: "ZX128_0.ROM" (16Kb) (2nd internal default)

AND the 16Kb file from the "ROMPAGE2" argument, if supplied,

or "ZX128_1.ROM" (16Kb) (internal default)

The Spectrum 128K+2 ROM(s):

Either a single 32Kb file or TWO 16Kb files, as follows:

If supplied: The 32Kb file from the "ROMPAGE3" argument.

Else: "Plus2.rom" (32Kb) (1st internal default)

Alternatively:

If supplied: The 16Kb file from the "ROMPAGE3" argument.

Else: "ZXP2_0.ROM" (16Kb) (2nd internal default)

AND the 16Kb file from the "ROMPAGE4" argument, if supplied,

or "ZXP2_1.ROM" (16Kb) (internal default)

Confused? :-) Then just use the ROM pack available from [my website](#) , or supplied in some versions of the archive, and do not supply any ROMPAGE n arguments.

These files can be, by default, placed in the same directory as the ASp executable. You can use the ROMPATH tooltype to specify a different path to use for the ROM files. A fully qualified path and file definition for any ROMPAGE n tooltype will override the ROMPATH for that file only.

The ROMPATH and ROMPAGE n arguments can be supplied using [Tooltypes](#) , or you can use the [ROM Configuration GUI](#) to modify which ROMs to use once ASp is running.

ASp must find suitable files for the ROM images before it will successfully initialise.

ASp will load and use any files of the correct name and length. Use of files which are not real Spectrum ROM images should not cause any harm, but will very likely prevent any useful functionality.

1.11 ASp Quick Start Guide

For those who don't want to read all the blurb before running ASp:

Make sure you've got all the [Requirements](#) and have de-archived the ASp distribution somewhere. Double-click the ASp icon to begin.

You should see a small ASp GUI window containing several buttons appear on your default public screen. If an error requester appears, you will have to correct the problem before proceeding.

The emulation is controlled by the RUN and STOP buttons. You can also press the Escape key to Stop the emulator.

Drag and drop files over the ASp window to load them. Alternatively, "Load" will load a snapshot, or "Insert" will open a tape file for reading. You can use Z80, SNA, SP, TAP and TZX files.

You can press RESET at any time to do a power-on reset of the emulated Spectrum.

The Amiga keyboard behaves exactly as would a Spectrum keyboard when ASp is running. To access the extended Spectrum BASIC commands use Shift as "Caps Shift" and Alt as "Symbol Shift". The Ctrl key will get to "Extended Shift" mode, as will pressing both Shift and Alt like a real Spectrum. Several shortcuts are available including the cursor keys, backspace, and some symbol keys.

Be aware that some Amiga keyboard controllers have problems with multiple simultaneous key presses. This is only really a problem for games. If the game has no joystick option then it is suggested you investigate the [Custom-Keys joystick](#) .

A standard digital joystick plugged into the normal Amiga GamePort will behave as a Spectrum joystick of the [type](#) set in the Settings menu. Default is Kempston.

A range of **audio** and **video** options are available which will affect the accuracy of the emulation in exchange for potential speed improvements.

To quit ASp, STOP the emulation then close the main window or select the Quit menu item.

Bug reports and feature requests are encouraged, but please read the detailed section(s) of this guide first.

1.12 What features are coming up?

Just because ASp doesn't currently have a feature does not mean it has never occurred to me! :) Indeed, I have had all sorts of plans for ASp, some of which I've rejected on the grounds of time, information, practicality, popularity, etc.

If you have an idea for a feature you'd like to see, or if you think there's things I shouldn't do / have done, **let me know** .

Here are some things I might implement at some time (no particular order)...

128K modes with no MMU.

Improve the status line.

TAP file saving.

Auto-stop the tape.

ROM Config. saved with prefs.

Improve tape noise.

Auto cheat-finder/poker.

Crunched saves.

Full SLT/DAT support.

Disk support.

+3 emulation.

Dual joysticks.

Dynamic JIT compiling emulation core.

More ARexx commands.

Internal Z80 debugger/disassembler.

A pop-up help screen for keyword position.

1.13 Detailed Usage Guide for ASp.

Please select the following links for information relating to that feature.

[Basic Operation](#)

[Keyboard usage](#)

[Joystick usage](#)

[Loading Snapshots](#)

[Saving Snapshots](#)

[Using Tape files](#)

[Emulated Machine Type selection](#)

[Speed Synchronisation selection](#)

[Video configuration options](#)

Audio configuration options

Miscellaneous configuration

ROM Configuration GUI

Load and Save Settings

ToolTypes & CLI arguments

ARexx port

Grab the display to a file

1.14 Basic Operation Overview

Main Emulation Control

On the left hand side of the main ASp window are three buttons, "RUN", "STOP" and "Reset". These are the primary controls of the emulation.

Pressing RUN will open the Spectrum display if necessary, and the emulated machine will run from where it was last stopped. If a snapshot has just been loaded, the program contained in the snapshot will begin to run.

Pressing STOP will pause the emulated machine. If the display was in Screen mode, the display screen will close. The ESCAPE key duplicates the function of the STOP button.

The RESET button will immediately simulate a power-on reset of the emulated machine. All unsaved data in the Spectrum will be lost.

These functions are also available in the Control menu.

Snapshots

The next group of buttons allow loading and saving snapshots of the emulated machine. Snapshots store the entire state of the emulated machine, so you can return to exactly the same task by re-loading a saved snapshot.

The "Load" button will allow you to specify a file to load. ASp will automatically switch between the 48K and 128K models to suit the requirements of the snapshot file. See the section on [loading snapshots](#) for more information.

The "Reload" button will instantly revert to the state immediately after the last snapshot load.

The "Save" button will allow you to save the state of the emulated Spectrum to a snapshot file for later retrieval or use with another program. See the section on [saving snapshots](#) for more information

ASp will load and save snapshot files of type [.Z80](#) , [.SNA](#) (48K type), and [.SP](#) .

These functions are also available in the Project menu.

Tape Images

The rightmost group of buttons are for use with tape images. These files differ from snapshots insofar as they contain data that the emulated Spectrum may use, but do not influence the current state of the emulator or emulated machine.

The "Insert" function will allow you to select a file to insert into the virtual tape deck. The Spectrum will then be able to read data from this "tape" through the normal access mechanisms, as if a real tape deck were attached. You can use [TAP](#) or [TZX](#) files.

You can "Eject" the tape if you wish, once you have finished with it.

The "Browse" function will open the [tape browser](#) window, which gives a visual display of the content of the tape and where it is wound to.

ASp can currently only read tape files. It cannot write to them.

Tape control is also available as items in the Control->Tape submenu.

Quitting ASp

ASp can be [quit](#) by closing its window or selecting the Project->Quit menu item.

1.15 How to Quit ASp

To exit from ASp, you can select the Project->Quit menu item or simply close the GUI window.

ASp will not Quit if the emulation is currently Running, unless the ARexx "QUIT FORCE" command is issued.

If ARexx scripts have been invoked by ASp using the Project->Run ARexx... menu item which have not yet completed, ASp will report a status of "Waiting..." until all scripts have terminated, after which it will exit.

1.16 Keyboard layout information

When ASp is running, the keyboard behaves like a Spectrum keyboard, not an Amiga keyboard. Some keypresses will not do quite what is expected unless you are familiar with Spectrum operation.

Unshifted letter and number keys work as expected, as do Space and Enter.

Spectrum has two shift keys, Caps-Shift and Symbol-Shift. In ASp, Caps-Shift is activated by pressing either of the normal Shift keys, Symbol-Shift is available by using either of the Alt keys. To access the Spectrum's "extended shift" mode, press Shift and Alt simultaneously, just like on a real Spectrum where you'd press Caps-Shift and Symbol-Shift. Alternatively, the Ctrl key will simulate pressing both buttons.

The symbols printed on the Amiga keyboard above the number keys cannot be accessed directly by pressing shift-1,2,3 etc. The Spectrum assigns other functions to shifted number keys, the symbols are available elsewhere on the Spectrum keymap using Symbol-Shift.

Some symbols and other functions usually accessed by two or more keystrokes on a real Spectrum are available as single-key shortcuts from the Amiga keyboard. The comma (,), full stop (.), slash (/), semicolon (;), hash (#), minus (-), and equals (=) are from the corresponding keys near the "return" key. The numeric pad also operates as labelled, with numbers, enter, "-", "+", "(", ")", "/" and "*" operating the corresponding Spectrum functions.

Backspace is a shortcut for Caps-Shift-0 (delete) and the cursor keys operate as Caps-Shift-5,6,7,8 for Spectrum cursor action.

Some keys, like Tab, Caps Lock, "~", etc, have no function within ASp.

Note that the above combinations are intended for use from within the Spectrum BASIC environment, simulating the Spectrum keyboard as marked. Some pieces of software may not work with these shortcuts.

Due to certain limitations, the keyboard layout corresponds to a UK Amiga keymap, it is not affected by locale settings. If this really causes people major problems there may be a way round this, albeit restricted.

Some F-Keys perform GUI shortcuts, but can be disabled via [ToolTypes](#) .

F1 is Load,

F3 is Save,

F6 is ReLoad,

F7 is Tape Start/Stop,

F9 is Reset.

You can use the F-Key shortcuts at any time.

In some instances, usually games, you may need to press several keys simultaneously. Some Amiga keyboard controllers (notably the A1200) do not recognise certain combinations of multiple keypresses. This is a limitation of the keyboard controller, not of ASp.

To circumvent this problem, define your keys such that they do not share the same rows of the keyboard. Try, for example, 3,E,K,L,Space for "Up, Down, Left, Right, Fire" respectively. Alternatively, investigate the use of the [Custom-Keys Joystick](#) mode.

1.17 Types of Joystick available in ASp

ASp offers a choice of Spectrum joystick types. Select which type of joystick you want fitted to your emulated Spectrum from the Settings->Joystick menu. The normal Amiga joystick will then appear to the Spectrum to be that type of joystick. You can change the joystick type at any time.

Types of joystick available:

Kempston (normal) (Default)

Kempston (smart)

Cursor

Sinclair 1

Sinclair 2

Custom Keys

As of V0.82, ASp allows user-selection of method used to access the Amiga game port. Try adjusting these if you have compatibility problems.

The following options are available:

Direct Hardware Stick

Gameport.dev Stick

Lowlevel.lib Auto

Lowlevel.lib Pad

Lowlevel.lib Stick

Also see **Keep Kempston Interface** menu item.

1.18 Kempston Joystick operation

Activated by selecting the "Settings->Joystick->Kempston (normal)" menu item.

Kempston is by far the most popular joystick interface type used by the Spectrum. Almost every program will work with this type of stick.

ASp will emulate a straightforward Kempston interface which is read by IN'ing from any port with address lines A5,A6 and A7 logic low (typically port 31).

See also: **Smart Kempston** joystick type.

1.19 Smart Kempston Joystick operation

Activated by selecting the "Settings->Joystick->Kempston (smart)" menu item.

Behaves as the **Normal Kempston** joystick type with one additional feature:

Some games (seems to be especially Spanish ones) will use an unspecified joystick type which superficially appears to be Kempston, but the normal Kempston decode will not work with these games. Should such a game be running, identified by accessing ports with just A5 low, ASp will switch from treating this as a "normal" Kempston and instead respond with the Up and Down directions swapped.

It is advised that you use the normal Kempston unless you need this feature.

1.20 Cursor Joystick operation

Activated by selecting the "Settings->Joystick->Cursor" menu item.

This makes the Amiga joystick appear to the Spectrum to be a Cursor joystick. A Cursor interface emulated the keys 5,6,7,8, and 0 being pressed, corresponding to the Spectrum's (unshifted) cursor key locations.

1.21 Sinclair 1 Joystick operation

Activated by selecting the "Settings->Joystick->Sinclair1" menu item.

The Amiga joystick will appear to behave as Stick 1 (Right) of the Sinclair Interface 2. This emulates keypresses 6,7,8,9,0.

1.22 Sinclair 2 Joystick operation

Activated by selecting the "Settings->Joystick->Sinclair2" menu item.

The Amiga joystick will appear to behave as Stick 2 (Left) of the Sinclair Interface 2. This emulates keypresses 1,2,3,4,5.

1.23 Custom Keys Joystick operation

This mode allows you to define an arbitrary set of keys, corresponding to Amiga joystick or joypad controls.

To define the keys, select the "Settings->Joystick->Custom Keys..." menu item. A window will appear presenting you with the 11 current key-bindings. A cursor is positioned over the topmost entry. Pressing a valid Spectrum key will assign that key to the corresponding joypad function and move onto the next assignment field. TAB will skip over the assignment, leaving the original value.

The key-binding selected for each key is displayed as the letter or number for which the Spectrum interprets that key. In addition, special symbols are used for non-alphanumeric keys: "sp"=space, "en"=enter, "cs"=caps-shift, "ss"=symbol-shift. Keys on the Amiga keyboard that correspond to multiple Spectrum keypresses are not allowed.

When you are happy with the keys specified, click "Accept". You do not have to specify all assignments. Some **joystick access methods** in conjunction with some joysticks will allow use of "fire 2" (blue) and "fire 3" (yellow) in addition to the normal up/down/left/right/fire functions. The remaining functions are only accessible when using a CD32 pad and lowlevel.library.

When the Custom Keys joystick type is active, all the keybindings are active. Using the joystick or joypad makes the emulated Spectrum believe the corresponding keys are being pressed.

If another joystick type is active (eg. Kempston), then the Up, Down, Left, Right and Fire functions will correspond to the behaviour of that joystick type, not the keys. However, the remaining six key-bindings corresponding to other buttons on the joypad are always available and active, even when Custom Keys stick type is not selected.

In general, the use of the Custom Keys settings does not in any way affect the normal use of the Spectrum keyboard. However, trying to use the same key on the keyboard as you are "already pressing" on the joypad may lead to unexpected results.

You may change the key-bindings and/or joystick type at any time.

The Custom Keys Joystick is an extension of a device which was available for real Spectrum models. A joystick interface was available for which a short program could be loaded from tape to configure which keys would be emulated for joystick movement. The key-bindings were then emulated in hardware until the machine was powered down (or the config program reloaded).

1.24 Hardware Direct port access for Joysticks

Using this option causes ASp to directly read the Amiga custom hardware registers associated with the game port. This will only work on "Classic" Amiga hardware. It is recommended that this mode is not used unless the other access modes give problems.

This mode can only read the controls of a basic joystick (Up, Down, Left, Right, Fire), so will not be able to use the full set of keys associated with the CD32 gamepad in the [Custom Keys](#) joystick options.

[Joystick usage index](#)

1.25 Auto-detect lowlevel.library port access

This option uses lowlevel.library to read the hardware attached to the Amiga gameport.

Lowlevel.library will attempt to auto-detect whether you have a joystick or CD32 gamepad attached, and behave appropriately.

[Joystick usage index](#)

1.26 Lowlevel.library port access for CD32 gamepad

This option uses lowlevel.library to read the hardware attached to the Amiga gameport.

Lowlevel.library will assume you have a CD32 gamepad attached. If you do not, unspecified behaviour may occur. Use this mode only if you cannot reliably access the pad in [System Auto](#) mode.

[Joystick usage index](#)

1.27 Lowlevel.library port access for joystick

This option uses lowlevel.library to read the hardware attached to the Amiga gameport.

Lowlevel.library will assume you have a joystick attached. If you do not, unspecified behaviour may occur. Use this mode only if you cannot reliably access the stick in [System Auto](#) mode.

[Joystick usage index](#)

1.28 Gameport.device port access for joystick

This option uses gameport.device to read the hardware attached to the Amiga gameport.

This is the preferred access method if you do not have lowlevel.library on your system. It can access up to three fire buttons on a joystick, but cannot access the CD32-specific green/fwd/rev/play functions.

Some versions of the lowlevel.library do not release the system resources after use. Therefore this mode may not operate if you have previously used lowlevel.library access modes on your system.

[Joystick usage index](#)

1.29 Emulation Speed adjustment

The Settings->Speed menu provides the facility to speed up or slow down the emulation relative to a real Spectrum. You can select from a range of speeds both slower and faster than normal, or a special "maximum" speed mode.

For the scaled percentage modes, the entire emulation runs at a reduced or accelerated speed, but still regulated (if your Amiga is sufficiently powerful) to a constant rate with smooth sound.

In the "MAX" mode no synchronisation is done at all, and no sound is available. The emulation runs at the fastest rate your computer will support at any given time, without any attempt to maintain a constant or regulated speed. Be aware that this can reach many times the normal speed.

These settings are NOT an adjustment of the Spectrum's Z80 CPU clock. The emulated machine will not see any difference internally. As a consequence, the frame update rate, the graphical FLASH timing and the pitch and timing of the sound will all change speed to suit. All "time delays", clocks, etc. will be scaled along with this setting.

The amount of Amiga CPU time required to maintain the emulation scales with the setting too. The "50% speed" mode runs the Spectrum at half speed and will only consume approximately half the Amiga CPU time compared to normal. The "200% speed" mode will attempt to run the emulation at twice full speed, and will only be able to do so if you have twice as much CPU power available than you would normally require. "MAX" mode will consume all available CPU time.

1.30 Loading Snapshots

A snapshot is, as the name suggests, a description of the instantaneous state of the emulated Spectrum. The purpose of such files is to be able to record the entire contents of the emulated Spectrum and what it was doing at the time such that it can be re-loaded at a later date. You can use this to save your work, your progress in a game, etc. The most common use, however, is to exchange ready-to-run images of programs with other users. Please read [this text](#) about the legalities of this action.

To load a snapshot, press the GUI Load button, or select Load from the Project menu. Alternatively, drag and drop the required file onto ASp's window.

For methods other than drag-and-drop, you will be presented with a file requester for you to select the file you wish to load. By default, only those files with names corresponding to snapshot formats are displayed. If you have a valid snapshot file which does not have such a name, modify or clear the "pattern" field of the requester to see it.

Double-click the desired file. ASp will attempt to load it. If it is unsuccessful, an error message will be displayed. In the event of an error the previous state of the emulation is unaffected.

ASp will currently load **.Z80** , **.SNA** or **.SP** snapshot files. Other formats may be added in the future. The **XFD package** will be used, if present, to transparently decompress snapshot files. This allows you to load files crunched with Powerpacker, Imploder, XPK, and most other Amiga crunch formats.

If there is an icon associated with the selected snapshot file and the **Settings->Misc.->Load snapshot prefs** menu item is active then ASp will scan the icon for **tooltypes** and will modify its settings to match. This allows you to maintain profiles of your favourite settings for any snapshot.

Any time after a successful Load you can press Reload, or select the Project->Reload menu item to return to the state the emulator was in immediately after loading the last snapshot. Note that icon tooltypes do NOT affect settings when doing a ReLoad.

1.31 Saving Snapshots

To save a snapshot, press the GUI Save button, or select Save from the Project menu.

You can perform a save at any time to store the instantaneous state of the emulation into a disk file for later retrieval or to load into other emulators.

When you select Save, you will be presented with a file requester. Type in the name of the file you wish the information to be saved into, then press the Save button in the requester. If the file already exists you will be asked to confirm that you wish to overwrite the file.

You should choose names which have ".z80", ".sna" or ".sp" appended to them for better identification. If you specify a file extension like this, ASp will save the file in that format. A filename specification without any extension will be saved as a **.Z80** . If you specify an unsupported extension, ASp will not save the file.

You are highly recommended to always use the **.Z80** format. It generates smaller filesizes than the **.SNA** or **.SP** format and can store information about the 128K modes.

If you try to save to a **.SNA** or **.SP** file when the emulator is in 128K mode, a warning requester will appear informing you that the format is not suitable. You can ignore this if you choose, but you are unlikely to be able to successfully retrieve your saved work when you re-load it unless you really know what you are doing!

1.32 Using tape files

A tape file describes the data stored on a Spectrum cassette tape. ASp uses **.TAP** or **.TZX** format.

To select a Tape file for use, click the Insert GUI button or select Insert from the Control->Tape submenu. Alternatively, drag and drop the file onto the ASp window.

When the file requester appears, double-click the tape file you wish to use, or press Insert in the requester. The selected tape image is inserted in the emulated tape deck and "rewound" to the start. ASp will decrunch any compressed files using the **XFD package**

The next time the emulation attempts to load from tape the Spectrum will be fooled into thinking it is reading information from a real tape. You can Eject it, or Insert a new one whenever you want.

Select "Browse" to study the content of the tape in the **Tape Browser** .

ASp can only read, not write to tape files.

There is an option, **Flash Load Standard Blocks** , which can help speed up the loading process in some cases.

1.33 Project->Grab Display...

This function can be selected at any time to capture the current Spectrum display to an IFF image file. The display is saved as-is, including rainbow effects. Also see the options to grab the **border** and **compress** the file.

The palette and colour detail of the image saved are internally generated and are not dependent on the current display mode, pen allocations, etc.

1.34 Description of .Z80 files

The ".Z80" snapshot file format was devised by Gerton Lunter for use with his emulator for PC's called, funnily enough, "Z80"...

It is a comprehensive format which contains information about all the features of the 48K and 128K Spectrums and additional hardware. File size is reduced significantly in many cases by using a simple run-length compression algorithm.

These files are extremely popular for snapshots on all platforms, and it is recommended that ASp users try to use this format for any saves made. This is the only format supported by ASp that can fully describe the 128K modes correctly.

There are three basic versions of .Z80 files. ASp should load any of them.

ASp saves version 2 .Z80 files. Most emulators with limited support for .Z80 files will deal with this version. ASp saves them with optional run-length compression active.

1.35 Description of .SNA files

The ".SNA" (also ".snapshot" or ".Mirage") file is a simple description of the state of the 48K Spectrum and a full RAM dump based on the Mirage Microdrive save format.

SNA files are always 49179 bytes in length and contain only enough information to describe the 48K Spectrum status. This makes them unsuitable for use as a means of snapshotting the 128K Spectrum. There is an extended SNA128 format which does contain additional information, but is rare and ASp does not support it.

SNA files are often found compressed on the Amiga. ASp V0.78 and above uses the **XFD package** to decrunch files from almost any packer.

1.36 Description of .SP files

The ".SP" format is suitable for making 48K Spectrum snapshots, originally used with the VGASpec emulator. It does not contain enough information to snapshot a 128K machine.

SP files are always 49190 bytes long (or 16422 for 16K Spectrum), slightly larger than the functionally similar **SNA** format. Although considerably rarer than SNA files, this format is somewhat superior insofar as the initial Program Counter is specified literally, rather than the SNA system of it being available on the stack, potentially trashing data.

ASp will load and save this format, but it is recommended to use the **Z80** format for saves.

1.37 Description of .TAP files

There are two types of .TAP file for the Spectrum. The one ASp supports is the more popular type used by the "Z80" emulator by Gerton Lunter. The other type is for an emulator called "Warajevo" by Samir Ribic. ASp does not support those.

A TAP file is a literal translation of the data on a Spectrum cassette tape, in the order it appears on the tape. All of the header and checksumming information used by the Spectrum is also there.

The TAP format is only suitable for use with programs using tape data in the same form and timing as that used by the standard ROM load and save routines. There are many programs which use their own custom tape loading routines and indeed their own private tape data formats. These tapes are normally saved for emulator use in another tape format called **TZX**. Some TAP files, however, contain data which is designed to be loaded via such a custom routine.

1.38 Description of .TZX files

The TZX file format is designed to allow representation of any tape loading format or sound. This is achieved by describing the tape in terms of tones and pulses or, where possible, predefined popular data patterns, such as those used by the ROM loader.

A TZX file can also contain other information, such as instructions, cheats, trainers, comments, etc. [Here](#) is a list of those TZX features supported by ASp.

1.39 TZX Blocks supported by ASp

ASp does not presently handle all the features of the full TZX specification. Most TZX files will be usable, however, and ASp should still be able to handle files which contain data it does not yet support.

All standard Spectrum-related blocks defined in the V1.13 TZX specification are recognised. The following are fully supported:

Block ID Description

\$10 Standard ROM-timed block

\$11 Standard custom-timed block

\$12 Pure tone

\$13 Sequence of Pulses

\$14 Custom-timed data-only block

\$20 Pause/Stop

\$23 Jump

\$24 Loop Start

\$25 Loop End

\$26 Call

\$27 Return

\$28 Select choice

\$2A Stop tape if 48K mode

\$30 Text comment

\$32 Archive Info

\$5A Extension block

The following are partially supported:

Block ID Description Limitations

\$21 Group Start Groups cannot be "collapsed"

\$22 Group End in the browser.

\$35 Custom Information Subtypes "Instructions" and "Picture" are supported.

The following are unsupported (do-nothing) blocks:

Block ID Description

\$15 Direct Recording block.

\$31 Timed Message

\$33 Hardware Compatibility Info

\$34 Emulator-Specific Info

\$40 Embedded Snapshot

1.40 Emulated Machine Type Selection

The Settings->Machine sub-menu provides the user with the ability to switch between the different supported Spectrum models.

Simply use the menu to select which of the available models you want ASp to emulate. Currently available are:

Sinclair 48K: The standard 48K Spectrum, Issue 2 or 3.

Sinclair 128K: The first 128K Spectrum, with AY sound chip.

Amstrad 128K+2: The revised 128K Spectrum released shortly after

Amstrad acquired Sinclair's copyrights.

In many situations, changing the model type in this menu will cause the emulated machine to "crash", so the standard procedure is for ASp to perform an automatic power-on reset when changing the selection. The Reset on machine change option controls this behaviour. Disabling this option will allow you to change machine type without affecting the running program. Only certain programs at certain points will safely accept a "live" machine change.

When you Load a snapshot, ASp will automatically select either 48K or 128K mode dependent on the information contained in the snapshot file. This completely overrides any previous user selection.

Use the menu item "Make 48K Issue 2?" to cause the 48K model to behave as an Issue 2. Otherwise, Issue 3 behaviour is standard. Very few pieces of software are affected by this switch, except for some very old games. For best operation, leave as Issue 3 unless a problem arises.

1.41 Audio Configuration

There are several options to configure ASp's audio output to your preferences.

[AHI Mode selection](#)

[Channel Mode selection](#)

[AY Volume Characteristic](#)

[Sample buffer size](#)

[Swap stereo channels](#)

[Quiet 48K beeps](#)

[Audible Save mode](#)

All these options can be changed "on the fly".

1.42 Settings->Audio->Select AHI Mode

Choose an AHI Audio Mode and sample rate from those available on your system.

A list of modes from the AHI audio mode database will be presented. Only those modes suitable for use with ASp will be shown. By default, ASp uses the mode defined in the AHI preferences program for "music unit".

It is recommended you select a mode which has at least 10 bits of resolution.

Some Amiga configurations may swap the left/right channels when using the Paula audio driver. The `AHIpaulaSwapChannels` environment variable can be set to "1" to control this. Please read the AHI documentation for more details. Also look at the [Swap stereo channels](#) option.

You must also choose the sample rate. Higher sample rates will offer clearer, cleaner sound, but will consume extra CPU time to process. The majority of sounds used in Spectrum software will not significantly improve above about 24kHz. It is recommended to select at least 8kHz.

The actual sample rate used by ASp may vary slightly from that selected.

Back to [Audio configuration options](#)

1.43 Audio Channel Mode

The channel mode is set from the Settings->Audio submenu.

This selection affects the sound stage, in particular the positioning of the AY sound chip channels.

In No Audio mode, ASp does not allocate any audio resources, and therefore makes no sounds. Timing synchronisation is carried out using `timer.device`. This mode may offer a small speed increase for some users since no sound processing is done at all.

In Mono mode, all of the emulated sounds, 48K and AY chip, are mixed into one stereo-centre channel.

In the Stereo modes, the 48K sounds are mixed in as "stereo centre", while the AY chip sounds are positioned according to the selection:

"Stereo ABC" plays AY channel A left, B centre and C right.

"Stereo ACB" plays AY channel A left, C centre and B right.

None of the standard issue Spectrum 128K models had stereo sound output, which is best simulated by "mono" mode. However, some of the Spectrum "clones" did have stereo output, so a lot of AY-chip music software positions the sounds in a stereo sound stage. The most typical arrangement is "Stereo ACB", which is the default.

See the section about selecting the [AHI Mode](#) for information concerning possible stereo sound stage problems.

Back to [Audio configuration options](#)

1.44 Audio Buffer Size

There are limitations with the way AHI interacts with some types of audio hardware. Many soundcards will require a larger buffer of samples than, for example, the native Amiga Paula audio. To accomodate this, you can select whether to have a small, medium or large buffer size.

If you find the sound with ASp is breaking up badly, yet you are sure you have enough CPU time available to run ASp at full speed, you can try increasing the audio buffer size.

The problem with larger buffer sizes is the delay between the action that causes a sound and the sound itself being heard. For this reason, it is recommended you keep the buffer size as small as will work with your configuration.

In general, the higher the sample rate you select in the **AHI Mode** requester, the smaller the buffer size can be in ASp.

Back to [Audio configuration options](#)

1.45 Settings->Audio->AY Volume steps

The characteristic of the AY chip volume can be modified to best suit your listening environment.

Linear mode makes each volume step exactly the same. This tends to make volume changes less noticable.

Logarithmic 2dB per step mode is a good likeness of a real AY chip, but does not go quite as quiet. Use this mode if some of the sounds in 3dB mode are too quiet to hear.

Logarithmic 3dB per step mode is a theoretically ideal model of the AY volume characteristic. The lowest volume output in this mode is very quiet, and may be inaudible if not using headphones or a decent sound system. It is likely that 8-bit resolution audio hardware will struggle to resolve the quieter sounds.

Back to [Audio configuration options](#)

1.46 Settings->Audio->Quiet 48K Beeps

A real 128K Spectrum mixed the 48K sound with the 128K sound such that the 48K sound was as loud as all AY sound channels put together. By default, ASp follows this behaviour.

Activating the Quiet 48K beeps option will cause ASp to mix the 48K sound only as loud as one AY channel, effectively increasing the volume of the AY chip sounds and decreasing the volume of the 48K sounds.

Back to [Audio configuration options](#)

1.47 Settings->Audio->Swap Stereo Channels

Swaps the left and right channels when in stereo modes.

Normally not required, but may be useful if there is a conflict between AHI sound stage vs. your sound system.

If using native Paula audio, also see the information about AHIPaulaSwapChannels in the **Mode Selection** section.

Back to [Audio configuration options](#)

1.48 Audio Save option

Menu item "Settings->Audio->Audible Save mode?"

This is a simple addition which sources the 48K beeper sound from the information sent to the MIC socket of the Spectrum/tape-deck as opposed to the normal EAR socket.

The result is that most program's attempts to SAVE to tape will result in an audible tape noise in ASp. In principle, this can be used to route into a real Spectrum for Loading. The normal 48K audio noises will often no longer work in this mode, so you should not activate this option unless you need to.

Note that it is vital that ASp is running at full speed without interruption of any kind to ensure the sound is clean and stable enough to be acceptable to the destination Spectrum.

This facility has not been fully tested, but is known to work. Users' reports of the degree of success in the use of this option would be appreciated.

Back to [Audio configuration options](#)

1.49 Video configuration options

These menu functions allow you to configure some aspects of ASp's video display.

Display Submenu:

[Screen or Window display](#)

Screen Submenu:

[Screen Mode selection](#)

[Screen centring](#)

[Colour Zero Border](#)

[Blank pointer](#)

[Force Chipmem Screen](#)

Window Submenu:

[Remember Position](#)

[Snap to Grid](#)

FrameSkip menu:

[FrameSkip setting](#)

Any of these options can be changed on-the-fly.

Additional Info:

[Automatic Calibration](#) of rendering functions.

[Palette selection](#) in window mode.

1.50 Emulation Display type selection

Settings->Display Type submenu

You can select either Screen or Window display.

When the Screen display (default) is selected, ASp will open a system screen of the mode and depth specified in the [Screen Mode selection](#). The emulated Spectrum display will appear on this screen which can be depth-arranged and dragged in the manner of any other system screen. When you Stop the emulation, the display closes.

Selecting the Window mode causes ASp to open a window on the same screen as the GUI window. Depending on the available palette, the colour accuracy of the display will vary. ([More about palette selection](#).) Stopping the emulation in Window mode will not cause the display to close, but the window can be closed at any time by using the window Close gadget.

When using window mode, especially with AGA or other native chipset, you may find a substantial performance increase by aligning the display to an internally calculated "grid", either by pressing the window "Zip" gadget or by activating the "Settings->Window->Snap to grid" menu option. ([Read more](#).)

Back to [Video Configuration options](#).

1.51 More information about palette selection.

When using the Window display mode, ASp attempts to allocate shared pens for its emulation display. Each time the window is re-opened, the allocation takes place again. ASp tries to allocate 15 pens.

On deep screens with many free pens, ASp will typically manage to acquire colours which are exactly the shade required. As fewer pens become available, ASp will obtain worse colour matches. In extreme cases the colours used may be completely unsuitable, in which case you should use the Screen display mode. Typically, attempting to use the Window mode with screens of depth 4 or less that have few free pens will result in unacceptable displays.

Technical information:

ASp acquires the pens using the system function ObtainBestPenA(). The algorithm used by this function is poor, especially with screens which are shallow (less than about 5 bit) or with few free pens. ASp therefore uses a series of iterative calls to attempt to maximise the overall colour selection performance.

It is, in principle, possible to patch the ObtainBestPenA() function to perform better. Unfortunately, the conflicting requirements of backward compatibility and greater performance mean that this is not straightforward and may require an unofficial extension to the function's parameters. Should such a patch become available, I will endeavour to make ASp compatible with it to maximise the authenticity of the display.

1.52 Screen Mode selection

Settings->Screen->Select screen mode...

This function will display a standard Screenmode requester. You can select any of the displayed modes for use with ASp. The available listed modes are dependent on your system configuration.

You can also select a number of colours (depth) from 16 to 256. It is advised that AGA users select the default 16 colours (4-bit) for maximum speed. RTG users may find that 256 colours (8-bit) offers better performance.

When ASp next opens the emulation display, it will try to open a screen of this mode and depth. If for any reason the screen is unable to open, ASp will display an error.

Regardless of mode chosen, the opened screen will be the same pixel dimensions. This means that if you pick a high-resolution screen mode you will get a very small emulation display. ASp will not scale the image to fit the screen size.

Nominally, ASp requires a display of 320 pixels wide and 256 pixels high with a 1:1 aspect ratio (square pixels) and a minimum depth of 4-bit.

The default mode is obtained by a call to BestModeID() for a full-screen display.

Back to [Video Configuration options](#).

1.53 Screen Centring options

Settings->Screen->Centre Horizontal?

Settings->Screen->Centre Vertical?

These options control the positioning of the emulation display when using the system screen display.

The default values for a new screen typically result in the screen being placed at the top-left of your monitor. If the resolution you have chosen is not a good fit to the full size of the monitor screen, you may prefer to have ASp attempt to centre the image on the monitor.

Centre Horizontal will cause ASp to compute an initial position to open the screen which will leave equal gaps to the left and right of the main display.

Centre Vertical will open a screen which is taller than the required size, such that the resultant position of the main display has equal gaps above and below it.

In both cases, the screen can still be dragged with the mouse to an alternative position once it is opened.

Note that the effectiveness of these functions depends on whether your Workbench display preferences have been correctly set. For graphics card users, your RTG software or graphics hardware may have limitations which prevent these options from functioning correctly.

Back to [Video Configuration options](#).

1.54 Colour Zero Border option

Settings->Screen->Colour Zero border?

ASp renders a 32-pixel border around the main display. This border supports "rainbow" effects when required. Often, however, the border is a static colour and a more authentic display is achieved by extending this colour off all edges of the screen.

Activating this option causes ASp to modify colour zero to whatever the "current" border colour is on a per-frame basis. For static borders on palette-mapped screens this offers an "entire-screen" Spectrum border. This option may not work with true- or hi-colour screen modes.

If rainbow effects occur in the border region, you will still see them in the rendered (inner 32-pixel) border, but the outer colour will not follow the rainbow effect.

If you have enabled the "border blanking" features of your graphics chipset, you will not see this feature since your colour zero is always black outside the main display area.

Back to [Video Configuration options](#).

1.55 Force Chip-mem Screen option

Settings->Screen->Force chipmem screen?

An option mostly for development use, but left in case it is useful to someone.

There is a system patch developed by Stephen Brooks and Przemyslaw Gruchala, called FScreen, that can allow the use of fastmem screens with the native chipset using a periodic refresh. This can offer considerable speed increase to applications, with some minor compatibility caveats.

Normally ASp will open its emulation display screen using standard system calls that are entirely compatible with, and independent of, whether you are using the native chipset or a graphics card. This is the normal, greatly preferable mode of operation, and is completely compatible to FScreen and virtually every other system patch.

However, due to the nature of the periodic refresh of FScreen, some Spectrum software with 50fps graphics effects can have (harmless) visual artifacts. ASp offers the "Force Chipmem Screen" option to open a deliberately non-standard display that is incompatible with, and thus not affected by, FScreen. This is achieved by forcing the display memory to be chip-mem.

Activating this option will permit users of FScreen to see 50fps Spectrum graphics as they were intended. Use of this option will, however, almost certainly cause serious problems if using RTG software. Since FScreen has no use with graphics cards this is normally not an issue. However:

DO NOT ENABLE THIS OPTION UNLESS YOU ARE USING FSCREEN AND A NATIVE CHIPSET (OCS/ECS/AGA) SCREEN MODE!

There are no speed benefits in activating this option.

This setting is not saved in the prefs.

Back to [Video Configuration options](#).

1.56 Blank Mouse Pointer option

Settings->Screen->Blank pointer?

When using Screen display mode the presence of the mouse pointer can be undesirable. This option allows you to turn it off.

Whenever the ASp display screen is "active" the mouse pointer will be invisible. The mouse functions still work - the screen can be dragged at the top, depth-arranged, etc. - but the pointer is not visible, so you will have to "guess" the mouse location.

Back to [Video Configuration options](#).

1.57 Remember display window position

Settings->Window->Remember position?

When using the Window display mode, this option affects where the emulation display window will open.

With this option enabled, ASp will open the display window wherever it was when you last closed it. Disabling this option causes ASp to try to open the window next to the main GUI control window.

Back to [Video Configuration options](#).

1.58 Window mode snap to grid.

Settings->Window->Snap to grid?

Activating this option will cause ASp to always move the display window such that the inner region is aligned to 32-pixels in the horizontal. Both the initial opening position and any subsequent manual movement will cause ASp to recalculate and move the window to remain aligned.

This only applies to Window mode, and is not required in Screen display mode.

[Read more information](#) about why you might use this feature.

Back to [Video Configuration options](#).

1.59 Why align the window to 32 pixels?

When ASp is using the Window display mode you can hit the Zip gadget to force the window to align to a 32-pixel boundary. Alternatively, you may have activated the "Snap to grid?" menu item, in which case the window is always aligned.

Why do this?

Rendering data into a display is most efficiently achieved if the source and destination memory is aligned to a 32-bit (4-byte) boundary. Any requirement for the data to be shifted left or right by some amount adds additional processing time and also creates more write cycles, slowing the rendering.

By making sure the edge of the Spectrum display is aligned to a 32-pixel boundary, it is assured that the data transfer is aligned to a 32-bit boundary and is therefore efficient. Depending on your graphics mode and chipset, it may actually only be necessary to align to 4 pixels, but the coarser alignment covers both this and the worst case condition.

For AGA users there is a substantial performance improvement by using the alignment features. Graphics card users may have varying results.

Note that it is the Spectrum display image which is aligned, not the edge of the window. System patches that incorrectly adjust the window border size information may cause this feature to break.

Back to [Video Configuration options](#).

1.60 Automatic Calibration of rendering routines.

As of V0.77, every time the display is opened, ASp will perform a benchmark of the system rendering calls which it uses. The time taken for this measurement is not significant.

ASp uses the measurements to assess the most efficient way to update the display. Depending on the depth and mode of the target screen, whether it is Screen or Window mode, and your specific system configuration, affects what the optimum update algorithm is. In particular, Cybergraphics calls will have been unnecessarily slow with the rendering system used in previous versions.

Back to [Video Configuration options](#).

1.61 Frame Skip function

ASp has to run the Spectrum's Z80 CPU emulation through every instruction, but even though a real Spectrum would update the video display every frame, there is no reason why an emulation has to do that.

The FrameSkip setting allows the entire beam-synchronisation (rainbow fx) calculations and display rendering update to be skipped for some frames. Since display-related computation can be as much as 50% of the CPU load, this can allow considerable extra processing power to be available for the Z80 CPU emulation, causing it to run faster. The disadvantage of using this function is the display will only be updated periodically, meaning smooth scrolling effects are no longer smooth.

The FrameSkip figure is the number of 50Hz Spectrum video frames which will occur without any updates. Hence, "FrameSkip 0" means NO frames will be skipped, all frames will be rendered. "FrameSkip 1" means one frame will be allowed to pass without rendering, meaning every other frame is rendered, etc.

The adaptive option, which is ON by default, will cause this behaviour to change slightly.

The selected FrameSkip figure becomes the maximum permitted FrameSkip, but ASp analyses how fast the emulation is running and will try to use a lower FrameSkip figure if it feels able to. In other words, you should obtain the best display update rate possible at any one time, subject to a worst-case limit set by your FrameSkip selection.

In most cases this is a more desirable behaviour, but with some software this can result in an inconsistent and distracting update frequency. Therefore the facility is provided as an option which is disabled by deselecting the Settings->FrameSkip->Adaptive menu item, causing ASp to use the fixed FrameSkip behaviour described above.

Back to [Video Configuration options](#).

1.62 Miscellaneous configuration options

The following settings relate to miscellaneous changes to the behaviour of the emulation.

[Add AY chip to 48K model](#)

[Keep Kempston Interface](#)

[Flash Load Standard Blocks](#)

[Delay During Flash Load](#)

[Enable F-keys](#)

[Load snapshot prefs](#)

[Grab display with border?](#)

[Compress IFF grabs?](#)

1.63 Settings->Misc->Add AY chip to 48K model

This menu item permits a slight "cheat" in terms of available hardware.

The AY sound chip was only found in 128K Spectrum models. However, several programs written for 48K machines were coded in such a way as to utilise this chip if it were present (extra sound, better title music, etc.), in case it was loaded into a 128K Spectrum.

ASp can, by activating this menu item, make the AY chip emulation respond even when in 48K mode. This enables the user to, for example, load a 48K-mode snapshot which is capable of playing 128K music.

In order for the user to hear the 48K-only music in a program which has either 128K or 48K music, the user must ensure this setting is inactive and that the emulation is running in 48K mode.

1.64 Settings->Misc->Keep Kempston Interface

This setting is a compatibility control. It is active by default.

Some snapshots have been saved at a point where the program has already decided that a **Kempston Joystick** interface is fitted. If you have selected a different joystick type for use or are using keys, so the Kempston Interface is not present, some programs can behave oddly.

This option causes a Kempston Joystick Interface to be plugged in at all times, regardless of the current joystick selection. Of course, if the current selected joystick type is not Kempston, the interface will not respond to joystick movement.

1.65 Settings->Misc->Flash-Load Standard Blocks

To aid with speeding up the **tape-loading** process, this option allows ASp to instantaneously "read" certain parts of the tape.

Any tape access using the standard ROM loader routine will be able to instantly load "Standard ROM-Timed Blocks". A **.TAP** file comprises exclusively of these blocks. A **.TZX** file may contain some of these, described in the **tape browser** as "STD BLOCK" or "STD HEADER".

Only the ROM-loader routine, using only those type of blocks, will be able to flash-load.

Note that enabling this option will cause the tape to automatically start playing when required.

Also see **Delay during Flash-Load** .

1.66 Settings->Misc->Delay during Flash-Load

When loading from a tape, ASp is able to **Flash-Load Standard Blocks** .

Activating the "Delay during Flash-Load" option will cause ASp to insert a brief delay after flash-loading each block. This allows loading screens etc. to be visible for longer.

1.67 Settings->Misc->Enable F-keys

Allows the use of the Function keys F1-F10 as shortcuts for certain often-used functions in ASp. The implemented functions are described in the **Keyboard** section.

Deactivating this option will cause ASp to completely ignore all function key presses.

1.68 Settings->Misc->Load snapshot prefs

When activated, causes ASp to look for an icon associated with a snapshot file when it is being **loaded**.

Tooltypes defined in the icon will cause ASp to adjust its settings to suit, after loading the snapshot. This provides a mechanism for the user to have preferred setups for each snapshot.

Use of the "ReLoad" function does not affect the settings.

1.69 Settings->Misc->Grab display with border?

When using the **Grab Display** function, this option controls whether the grabbed picture will include the Spectrum "border".

If this option is inactive, the resultant image will be 256x192 pixels from the main Spectrum display area.

If this option is active, the resultant image will be 320x256 pixels, which includes a 32-pixel border (including any rainbow effects), as displayed in ASp.

1.70 Settings->Misc->Compress IFF grabs?

When using the **Grab Display** function, this option enables the standard IFF "byterun" compression of the file.

There is very little point in deactivating this option unless the specific software you wish to load the image into does not accept compressed IFF images.

1.71 Loading and Saving User Prefs Settings

Settings->Load Settings

Settings->Save Settings

You can save the current user preference settings by selecting "Save Settings" from the Settings menu. All the options in the Settings menu are saved, including Screen Mode and Custom Key Joystick settings, along with the onscreen position of ASp's window(s), into a file called "ASp.prefs".

On start-up, ASp will attempt to load the "ASp.prefs" file and will use the configuration specified in it. You can manually revert to the saved settings at any time by selecting "Load Settings" from the Settings menu. Read **more** about ASp's start-up settings.

ASp does not save the name or path of any currently loaded snapshot or tape file, or whether the display is visible or the emulation is running.

From V0.84 the prefs file is similar to the tooltypes, so can be read and edited by humans. Do so at your own risk!

1.72 ASp initialisation behaviour

ASp can be loaded via the CLI or from Workbench by clicking on the icon associated with the ASp executable, or by clicking on a Project Icon associated with a snapshot where the default tool is ASp.

The initial user preference settings are derived in order as follows:

- The ASp.prefs file is used (if present) to define the base settings.
- The icon associated with the ASp executable (if present) is scanned for tooltypes and the base settings adjusted to suit.
- If loaded from a project icon, this is then scanned for tooltypes to selectively

override the tooltypes and prefs specified so far.

Any combination, or none, of the above preference specifications can be supplied. See the [tooltypes](#) section for details about specifying individual preferences.

If loaded from a project icon then ASp will load and run the associated snapshot file immediately upon start-up. When running ASp from the CLI, use the "SNAPSHOT" [argument](#) to specify an initial file to load and run.

NOTE: The load-and-run on startup only applies to snapshots, not tape files!

1.73 ASp ToolTypes

ASp supports a number of tooltypes to control its behaviour. These can be divided into two groups: those which define one-off initialisation parameters and those which control preference settings changeable by the user during program execution.

Tooltypes may be of type [LITERAL](#) , [BOOLEAN](#) , [STRING](#) or [NUMERIC](#) . Please select the links to find out how to specify each type.

Tooltypes with invalid arguments will be ignored.

The following are initialisation-only tooltypes. These are observed only during [start-up](#) and have no effect when loading snapshots with the [Load snapshot prefs](#) option activated.

[AREXXPORTNAME](#) (LITERAL)

[EXTIMAGEVIEWER](#) (LITERAL)

[EXTTEXTVIEWER](#) (LITERAL)

[LOADPATH](#) (LITERAL)

[ROMPAGE0](#) (LITERAL)

[ROMPAGE1](#) (LITERAL)

[ROMPAGE2](#) (LITERAL)

[ROMPAGE3](#) (LITERAL)

[ROMPAGE4](#) (LITERAL)

[ROMPATH](#) (LITERAL)

[SAVEPATH](#) (LITERAL)

The remaining tooltypes affect preferences which can be adjusted by the user during run-time by items in the Settings menu. These are observed both at [start-up](#) and when loading snapshots with the [Load snapshot prefs](#) option activated.

[48KISSUE2](#) (BOOLEAN)

[ADAPTIVEFS](#) (BOOLEAN)

[ADD48KAY](#) (BOOLEAN)

[AUIDBUFFER](#) (STRING)

[AUDIOMODE](#) (STRING)

[AYCHAR](#) (STRING)

[BLANKPOINTER](#) (BOOLEAN)

[CENTREX](#) (BOOLEAN)

[CENTREY](#) (BOOLEAN)

[COL0BORD](#) (BOOLEAN)

[DISPLAY](#) (STRING)

[FLASHLOAD](#) (BOOLEAN)

FRAMESKIP (NUMERIC)
JOYPORTACCESS (STRING)
JOYSTICK (STRING)
KEEPKEMPSTON (BOOLEAN)
MACHINE (STRING)
QUIET48 (BOOLEAN)
SCREENDEPTH (NUMERIC)
SCREENMODE (NUMERIC)
SPEED (STRING)
SWAPSTEREO (BOOLEAN)
TAPDELAY (BOOLEAN)
USEFKEYS (BOOLEAN)

1.74 Specification of type LITERAL tooltypes

Literal tooltypes take the entire of the argument as a single literal string. No case conversion or interpretation is performed.

Examples:

AREXXPORTNAME=MY_ASP_PORT

The argument "MY_ASP_PORT" is taken literally and used as the name.

ROMPATH=DH1:Misc/emulators/roms

The argument "DH1:Misc/emulators/roms" is taken literally and used as the filename.

1.75 Specification of type BOOLEAN

Boolean tooltypes interpret the argument as a yes/no switch.

There is a choice of syntax:

Use any of the following to activate an option: "ON", "YES", "Y", "1"

Use any of the following to deactivate an option: "OFF", "NO", "N", "0"

The arguments are not case-sensitive.

In addition, specifying the tooltype without arguments will be interpreted as "YES".

Examples:

USEFKEYS=Yes Activate

ADAPTIVEFS=0 Deactivate

BLANKPOINTER=N Deactivate

ADD48KAY Activate

1.76 Specification of type STRING tooltypes

String tooltypes interpret the argument as one of a series of possible options. See each tooltype description for details of which options are valid for each tooltype. The arguments are not case-sensitive.

Examples:

DISPLAY=WINDOW Set display type to window mode

JOYSTICK=Cursor Set joystick type to Cursor

1.77 Specification of type NUMERIC tooltypes

Numeric tooltypes interpret the argument as a number.

Normally the number is specified in decimal, but a leading "0x", "0X" or "\$" will cause the number to be taken as Hexadecimal.

Examples:

SCREENMODE=0x00021000 Set screenmode to hex 00021000

FRAMESKIP=5 Set frameskip value to 5

1.78 ASp Command-Line arguments

Invoking ASp with a snapshot file name as an argument will load and run that snapshot immediately upon start-up.

Examples:

ASp commando.sna

loads and runs "commando.sna".

1.79 AREXXPORTNAME

Valid only during start-up.

Format: AREXXPORTNAME=<name of arexx port>

The name of the ARExx port can be specified as any valid port name up to 32 characters in length. It must be a unique name otherwise ASp will display an error.

If not specified, ASp will attempt to derive a unique name itself.

[Tooltypes index](#)

1.80 EXTIMAGEVIEWER

Valid only during start-up.

Format: EXTIMAGEVIEWER=<filename of image viewer>

Allows the user to specify a viewer to use when ASp shows images, such as those in a TZX Custom-Picture block. It should be capable of viewing GIF and JPG files.

If not specified, defaults to "Multiview".

[Tooltypes index](#)

1.81 EXTTEXTVIEWER

Valid only during start-up.

Format: EXTTEXTVIEWER=<filename of text viewer>

Allows the user to specify a viewer to use when ASp displays lengthy text, such as that in a TZX Custom-Instructions block. It should be capable of displaying plain ASCII text in a monospaced font.

If not specified, defaults to "Multiview".

[Tooltypes index](#)

1.82 LOADPATH Argument

Valid only during start-up.

Format: LOADPATH=<initial path for requester>

Allows specification of which directory is initially displayed when the user first opens the Load or Insert file requesters.

If not specified, defaults to "PROGDIR:".

[Tooltypes index](#)

1.83 SAVEPATH Argument

Valid only during start-up.

Format: SAVEPATH=<initial path for requester>

Allows specification of which directory is initially displayed when the user first opens the Save file requester.

If not specified, defaults to "RAM:".

[Tooltypes index](#)

1.84 ROMPATH Argument

Valid only during start-up.

Format: ROMPATH=<directory containing ROM files>

Allows specification of a directory in which the Spectrum ROM image files can be found.

If not specified, defaults to "PROGDIR:"

Read the section on [which ROMs to use](#) for more information.

[Tooltypes index](#)

1.85 ROMPAGE0 Argument

Valid only during start-up.

Format: ROMPAGE0=<filename of ROM image>

Allows specification of a 16Kb image file of the Spectrum 48K ROM. The file is looked for in the directory optionally specified by the [ROMPATH](#) argument, but if you provide a fully-qualified path and file name here, it will be overridden for this file.

An 80Kb file containing all the ASp ROMs is also acceptable here.

Read the section on [which ROMs to use](#) for more information.

[Tooltypes index](#)

1.86 ROMPAGE1 Argument

Valid only during start-up.

Format: ROMPAGE1=<filename of ROM image>

Allows specification of a ROM image file containing either the 16Kb data of the first of the two Spectrum 128K ROMs, or the 32Kb of data of both Spectrum 128K ROMs. The file is looked for in the directory optionally specified by the [ROMPATH](#) argument, but if you provide a fully-qualified path and file name here, it will be overridden for this file.

Read the section on [which ROMs to use](#) for more information.

[Tooltypes index](#)

1.87 ROMPAGE2 Argument

Valid only during start-up.

Format: ROMPAGE2=<filename of ROM image>

Allows specification of a ROM image file containing the 16Kb data of the second of the two Spectrum 128K ROMs. The file is looked for in the directory optionally specified by the **ROMPATH** argument, but if you provide a fully-qualified path and file name here, it will be overridden for this file.

Read the section on **which ROMs to use** for more information.

[Tooltypes index](#)

1.88 ROMPAGE3 Argument

Valid only during start-up.

Format: ROMPAGE3=<filename of ROM image>

Allows specification of a ROM image file containing either the 16Kb data of the first of the two Spectrum 128K+2 ROMs, or the 32Kb of data of both Spectrum 128K+2 ROMs. The file is looked for in the directory optionally specified by the **ROMPATH** argument, but if you provide a fully-qualified path and file name here, it will be overridden for this file.

Read the section on **which ROMs to use** for more information.

[Tooltypes index](#)

1.89 ROMPAGE4 Argument

Valid only during start-up.

Format: ROMPAGE4=<filename of ROM image>

Allows specification of a ROM image file containing the 16Kb data of the second of the two Spectrum 128K+2 ROMs. The file is looked for in the directory optionally specified by the **ROMPATH** argument, but if you provide a fully-qualified path and file name here, it will be overridden for this file.

Read the section on **which ROMs to use** for more information.

[Tooltypes index](#)

1.90 48KISSUE2 Argument

Format: 48KISSUE2=<boolean>

Sets the state of the **Settings->Machine->Make 48K Issue 2** preference.

[Tooltypes index](#)

1.91 ADAPTIVEFS Argument

Format: ADAPTIVEFS=<boolean>

Sets the state of the **Settings->FrameSkip->Adaptive** preference.

[Tooltypes index](#)

1.92 ADD48KAY Argument

Format: ADD48KAY=<boolean>

Sets the state of the [Settings->Misc->Add AY chip to 48K model](#) preference.

[Tooltypes index](#)

1.93 AUBUFFER Argument

Format: AUBUFFER=<size>

Selects the audio [buffer size](#) preference.

Valid arguments are:

SMALL (20-40ms latency)

MEDIUM (60-80ms latency)

LARGE (140-160ms latency)

[Tooltypes index](#)

1.94 AUDIOMODE Argument

Format: AUDIOMODE=<mode>

Selects the [Audio Channel Mode](#) preference.

Valid arguments are:

NONE =No audio output

MONO =Mono

STEREOABC =Stereo, channel mode A-B-C

STEREOACB =Stereo, channel mode A-C-B

[Tooltypes index](#)

1.95 AYCHAR Argument

Format: AYCHAR=<characteristic>

Selects the [AY volume step characteristic](#) preference.

Valid arguments are:

LIN =Linear steps

2DB =Logarithmic, 2dB per step

3DB =Logarithmic, 3dB per step

[Tooltypes index](#)

1.96 BLANKPOINTER Argument

Format: BLANKPOINTER=<boolean>

Sets the state of the [Settings->Screen->Blank pointer](#) preference.

[Tooltypes index](#)

1.97 CENTREX Argument

Format: CENTREX=<boolean>

Sets the state of the [Settings->Screen->Centre Horizontal](#) preference.

[Tooltypes index](#)

1.98 CENTREY Argument

Format: CENTREY=<boolean>

Sets the state of the [Settings->Screen->Centre Vertical](#) preference.

[Tooltypes index](#)

1.99 COL0BORD Argument

Format: COL0BORD=<boolean>

Sets the state of the [Settings->Screen->Colour zero border](#) preference.

[Tooltypes index](#)

1.100 DISPLAY Argument

Format: DISPLAY=<display type>

Selects the [Display Type](#) preference.

Valid arguments are:

SCREEN =System-screen

WINDOW =Window-on-Workbench

[Tooltypes index](#)

1.101 FLASHLOAD Argument

Format: FLASHLOAD=<boolean>

Sets the state of the [Settings->Misc->Flash-Load Standard Blocks](#) preference.

[Tooltypes index](#)

1.102 FRAMESKIP Argument

Format: FRAMESKIP=<skipvalue>

Sets the [FrameSkip](#) preference.

Valid range is 0 to 5.

[Tooltypes index](#)

1.103 JOYPORTACCESS Argument

Format: JOYPORTACCESS=<access method>

Selects the active **joyport access** method.

Valid arguments are:

DIRECT Hardware-bashing joystick read.

AUTO Auto-detect stick or pad using lowlevel.library.

PAD CD32 pad using lowlevel.library.

STICK Joystick using lowlevel.library.

GAMEPORTDEV Joystick using gameport.device.

[Tooltypes index](#)

1.104 JOYSTICK Argument

Format: JOYSTICK=<joystick type>

Selects the active **joystick type** preference.

Valid arguments are:

KEMPSTON

SMARTKEMPSTON

CURSOR

SINCLAIR1

SINCLAIR2

CUSTOMKEYS

[Tooltypes index](#)

1.105 KEEPKEMPSTON Argument

Format: KEEPKEMPSTON=<boolean>

Sets the state of the **Settings->Misc->Keep Kempston Interface** preference.

[Tooltypes index](#)

1.106 MACHINE Argument

Format: MACHINE=<machine type>

Selects the **machine type** preference.

Valid arguments are:

48K =48K issue 3

128K =128K standard

128KP2 =128K Plus 2

You are advised not to use this tooltype in snapshot icons. If the "Reset on Machine Change" option is active, the emulation will be reset after loading the snapshot! Normally, ASp will automatically decide which mode is most suitable for a loaded snapshot.

Use as an initial start-up argument is not a problem.

[Tooltypes index](#)

1.107 QUIET48 Argument

Format: QUIET48=<boolean>

Sets the state of the [Settings->Audio->Quiet 48K beeps](#) preference.

[Tooltypes index](#)

1.108 SCREENDEPTH Argument

Format: SCREENDEPTH=<depth>

Selects the [screen depth](#) , in bits per pixel, preference.

Valid range is 4 to 32.

[Tooltypes index](#)

1.109 SCREENMODE Argument

Format: SCREENMODE=<mode number>

Selects the [screenmode](#) preference.

Example:

SCREENMODE=0x00021000 =PAL:Low-Res

[Tooltypes index](#)

1.110 SPEED Argument

Format: SPEED=<emulation speed>

Selects the [emulation speed](#) preference.

Valid arguments are:

50%

75%

100%

110%

125%

150%

200%

MAX

[Tooltypes index](#)

1.111 SWAPSTEREO Argument

Format: SWAPSTEREO=<boolean>

Sets the state of the [Settings->Audio->Swap Stereo Channels](#) preference.

[Tooltypes index](#)

1.112 TAPDELAY Argument

Format: TAPDELAY=<boolean>

Sets the state of the [Settings->Misc->Delay during TAP load](#) preference.

[Tooltypes index](#)

1.113 USEFKEYS Argument

Format: USEFKEYS=<boolean>

Sets the state of the [Settings->Misc->Use F-Keys](#) preference.

[Tooltypes index](#)

1.114 The ASp Tape file browser

By selecting Browse... from the GUI or the Control->Tape->Browse... menu item, you can open the tape browser. This offers a scrolling list describing all the blocks contained in the currently inserted tape file.

For [.TAP](#) files, the list will comprise entirely of blocks described as "STD HEADER" and "STD BLOCK". These are blocks of tape data which are formatted and timed in a manner compatible with the standard ROM loading routines.

For [.TZX](#) files, a variety of block descriptions may appear which show the structure of the TZX file. See [here](#) which of these ASp supports.

An arrow "=>" is drawn to the left of the "current block", the block to which the tape is currently wound. This defaults to the first block. To change the current block, double-click the appropriate list entry. Changing the current block (tape position) will stop the tape.

Next to some block descriptions is some information, such as the size of the data block, the name of the group, etc. For certain blocks, further information is available by pressing the "More..." button. See below.

The tape can be started/stopped at any time by pressing "Play/Pause" or F7.

When the Tape Browser is the active window, key presses will not operate the emulated keyboard. Instead, cursor up/down will move the highlighted tape block, the "enter" key will set the tape to the highlighted position (and stop the tape), "M" is the same as pressing the "More..." button, and "P" will start or stop the tape.

The browser window can be opened or closed at any time without affecting the tape or emulator activity. It is not necessary to have the browser window open to load from tape. (The tape can be started and stopped with F7.)

Additional information is available for certain block types:

Archive Info (Title, Author, Price, etc.) is presented in as much detail as is included in the file.

Instructions and Picture information in Custom Info Blocks will launch an asynchronous external viewer. Use [tooltypes](#) to change which viewer is used.

1.115 ASp ROM Configuration GUI

Selecting "ROM Config." from the settings menu brings up a window which allows you to set the names and write-protection status of the ROM files in use.

Each row represents one of the 16Kb ROM pages used by ASp. The first row is the Spectrum 48K ROM, the next two are the ROMs found in a 128K Spectrum, and the next two are for the Spectrum 128K+2. Each row contains a filename gadget, a "Rescan" button, and a "Write-Protected" checkmark.

The write-protection status of the ROMs can be changed on a per-page basis by the checkboxes. The default is always "protected", since correct operation of the Spectrum is not guaranteed if the ROM is write-enabled. Do not change these settings unless you have a specific reason to do so.

Suitable files can be specified by typing into the name fields. The filenames here follow the same rules as the **tooltypes** for ROM file specification. ie: Filenames are relative to ROMPATH unless fully qualified.

For each ROM page you can specify a file to load. If you simply want to re-load the file already specified, for example if you have changed the file contents or have had the ROM page unprotected, you may press "Rescan" to do this. Only those pages which are Rescanned or re-specified will overwrite the ROM data in use.

For additional flexibility and compatibility, some of the ROM page images can be combined into one file. Specifically, a single 80Kb (5x16Kb) file containing all ROM pages in the correct order can be loaded into the first row, or a 32Kb (2x16Kb) file containing both of the ROM pages for a 128K machine can be loaded into the first of the two rows relating to that machine. To avoid confusion, the filename boxes will be ghosted as they become occupied by larger files in earlier rows. You will get an error message if you attempt to load files of inappropriate or conflicting sizes into the rows.

The size of the files loaded into the rows does not affect the granularity of the page write-protection.

The changes made in this window do not take place until you press "Use". This button is not available until you have a valid ROM file configuration. Pressing "Cancel" will exit the window and discard any changes made.

1.116 Copyrights on Spectrum Software

It is a popular misconception that exchanging or downloading images of commercial software for use with emulators does not constitute software piracy.

Unfortunately, it IS piracy and IS illegal!

All is not lost, however! A growing number of software companies and authors have made public statements that specific titles for which they own the copyrights may be made freely available. In addition, the maintainers of **World Of Spectrum** website have taken a lot of time and effort to contact copyright holders to try to encourage them to allow free distribution. They have had a lot of positive responses, with many individuals giving them permission to retain copies for public download.

In all cases, the original author retains copyright! The software is simply being made free for use.

Please respect the generous gestures made by these individuals by not abusing the privilege of being able to use their software for free. Do not claim it as your own, do not distribute hacked or broken copies as originals. Always acknowledge their copyrights.

Please Note: The author of ASp is in no way responsible for the user's choice of program images used in conjunction with ASp. The use of illegal images of Spectrum software is entirely at the user's risk and discretion.

The World Of Spectrum website contains links to detailed information concerning the many viewpoints and legalities with this issue.

1.117 ARexx port functions

ASp creates an ARexx port for communication with other programs. By default, the port will be named "ASP_0" or, if that name is unavailable, "ASP_1", "ASP_2" and so on until it finds a unique name. Alternatively, the user can specify the name of the ARexx port to use by the **AREXXPORTNAME** tooltype or CLI argument.

ASp makes two uses of the ARexx port. **Executing commands** and **running scripts** .

1.118 Running an ARexx script from within ASp

To run an ARexx script, select the Project->Run ARexx... menu item. Select the name of the script you wish to run in the requester.

The ARexx program which is run from this function will address the ARexx port of the ASp copy it was called from by default, without having to determine its name.

You can use this feature as a kind of macro functionality, or to expand the number of features or functions ASp has.

Some demo scripts are supplied in the Rexx subdirectory. These are just very simple examples of what you could do.

[LoadSCR.asprx](#)

[SaveSCR.asprx](#)

[POKEr.asprx](#)

1.119 ASp ARexx Commands

ASp supports a selection of ARexx commands to enable macro functionality, or allow other programs and external scripts to control its features.

Important general information regarding ASp's command set.

Commands are likely to get added with subsequent versions, therefore the following list shows minimum version numbers of ASp which support the command. Commands new or modified in this version have the version number in bold.

V0.76 **HELP**

V0.76 **GETVERSION**

V0.75 **RUN**

V0.75 **STOP**

V0.75 **RESET**

V0.75 **QUIT**

V0.75 **LOAD**

V0.76 **SAVE**

V0.75 **RELOAD**

V0.76 **INSERT**

V0.76 **REWIND**

V0.76 **PEEK**

V0.75 **POKE**

V0.76 **READRAM**

V0.76 **WRITERAM**

V0.76 **GETPAGE**

V0.75 **SETPAGE**

1.120 ASp ARexx important information

ASp is an ARexx Command Host. The commands for which you expect a result will put the result in the "RESULT" ARexx special variable if, and only if, the ARexx "OPTIONS RESULTS" feature is switched on. You cannot treat the commands as functions. ie:

```
V=PEEK 16384
```

```
SAY 'Location 16384 contains 'V
```

is INCORRECT and will not work!

Whereas,

OPTIONS RESULTS

PEEK 16384

SAY 'Location 16384 contains 'RESULT

is CORRECT!

It is important to realise that the ARexx commands are asynchronous to the GUI. ie: If you STOP the emulation, the user could hit the RUN button on the GUI to start it again. Similarly, commands like POKE or PEEK will still execute while the emulation is running and therefore their meaning or usefulness may be limited if you do not first STOP the emulation.

It is up to the ARexx programmer to ensure his code is aware of the current state of the emulation when required and executes suitable interlock procedures to ensure that data passed to and from the emulation is meaningful.

1.121 ARexx command: HELP

Syntax: HELP

Arguments: None.

Result: Command summary.

Return code: OK

Function: Returns a short command summary text.

Unlike most other commands, this command will write the result to the command source's STDOUT if the ARexx OPTIONS RESULTS feature is not active.

1.122 ARexx command: GETVERSION

Syntax: GETVERSION

Arguments: None.

Result: Version number.

Return code: OK

Function: Returns a 4-digit decimal number which is the ASp executable version number multiplied by 1000. ie: V0.76 will return "0760".

In-between versions will increment the last digit.

Should be used to check the version of ASp is adequate to run the script, eg:

Options Results

GETVERSION

If RESULT>=0760 Then Do

...

1.123 ARexx command: RUN

Syntax: RUN

Arguments: None.

Result: None.

Return code: OK if emulation started ok.

WARN if emulation was already running.

FAIL if emulation could not be started.

Function: Attempts to open the emulation display and start the emulation as if the user had pressed the RUN button on the GUI.

1.124 ARexx command: STOP

Syntax: STOP

Arguments: None.

Result: None.

Return code: OK if emulation stopped ok.

WARN if emulation was not running.

Function: Stops the emulation and shuts down the emulation display, as if the user had pressed the STOP button on the GUI.

1.125 ARexx command: RESET

Syntax: RESET

Arguments: None.

Result: None.

Return code: OK

Function: Resets the emulation as if the user had pressed the RESET button on the GUI.

1.126 ARexx command: QUIT

Syntax: QUIT [FORCE]

Arguments: FORCE: Shut down emulation and don't ask user. (V0.76+)

Result: None.

Return code: OK if the Quit Requester was successfully displayed.

FAIL if the args were invalid, or:

if no "FORCE" argument, the emulation was running.

Function: Acts as if the user had selected the Project->Quit menu item.

Displays the "Quit? Yes/No" requester, which the user may cancel.

If the emulation was running, this command fails.

V0.76+ only:

Use of FORCE argument allows command to always cause shutdown even if running, and will not ask the user.

1.127 ARexx command: LOAD

Syntax: LOAD [<filename>]

Arguments: <filename>: Optional snapshot filename.

Result: None.

Return code: OK if file loaded ok.

ERROR if user cancelled filerequester.

FAIL if could not load file for any reason.

Function: Attempts to Load the snapshot file specified by <filename>. If no file was specified, displays the Load Snapshot requester.

1.128 ARexx command: SAVE

Syntax: SAVE [<filename>]

Arguments: <filename>: Optional snapshot filename.

Result: None.

Return code: OK if file saved ok

ERROR if the file requester was cancelled

FAIL if the file could not be saved

Function: Saves the state of the emulation to the specified file.

The format used is determined by the file extension. If none given, the Z80 format is used. Bad extensions will cause a fail.

If <filename> is not supplied, displays the Save Snapshot requester.

Versions prior to V0.82 did not behave correctly or consistently.

1.129 ARexx command: RELOAD

Syntax: RELOAD

Arguments: None.

Result: None.

Return code: OK if last file reloaded ok.

ERROR if no ReLoad buffer available. (ie: Previous load failed.)

Function: Recovers the state of the emulation after the last successful Load.

1.130 ARexx command: INSERT

Syntax: INSERT [<filename>]

Arguments: <filename>: Optional tape file name.

Result: None.

Return code: OK if file inserted ok.

FAIL if unable to read file.

Function: Inserts the tape file specified into the virtual tape deck. If no file is specified, the Insert tape requester is displayed.

1.131 ARexx command: REWIND

Syntax: REWIND

Arguments: None.

Result: None.

Return code: OK if tape rewound ok.

WARN if no tape currently inserted.

Function: Rewinds the currently selected tape file to the beginning.

1.132 ARexx command: POKE

Syntax: POKE <address> <value>

Arguments: <address>: The address (0-65535) to write to.

<value>: The value (0-255) to write to that address.

Result: None.

Return code: OK if value successfully written.

FAIL if <address> or <value> out of range.

Function: Writes the byte <value> to memory location <address>.

ROM locations cannot be changed by this command.

1.133 ARexx command: PEEK

Syntax: PEEK <address>

Arguments: <address>: The address (0-65535) to read from.

Result: Decimal value (0-255) of the contents of the memory location.

Return code: OK if value successfully read.

FAIL if <address> out of range.

Function: Reads a byte value (0-255) from memory location <address>.

1.134 ARexx command: READRAM

Syntax: READRAM <address> <length>

Arguments: <address>: Address (0-65535) of first location to read from.

<length>: Number of bytes to read. (1-16384)

Result: A data stream (string) containing byte values of each sequential location from <address> for <length> bytes.

Return code: OK if data read ok.

FAIL if <address> or <length> out of range.

Function: Reads a string of <length> bytes from <address> memory location onwards. <address> + <length> must not total more than 65536.

ie: Address wrap-around not supported.

1.135 ARexx command: WRITERAM

Syntax: WRITERAM <address> <data>

Arguments: <address>: The first address (0-65535) to write to.

<data>: A string of byte values to write to the location.

Result: None.

Return code: OK if values written ok.

FAIL if <address> out of range.

Function: Writes each byte value from <data> sequentially into memory locations starting at <address>. It is not possible to overwrite ROM with this command.

Address wrap-around is supported.

1.136 ARexx command: SETPAGE

Syntax: SETPAGE <value>

Arguments: <value>: Value (0-255) to send to port \$7FFD.

Result: None.

Return code: OK if output ok.

FAIL if <value> out of range.

Function: Allows control of 128K paging features by outputting <value> to port \$7FFD. The bits of this port have the following functions:

Bits: 0-2: RAM page number to map at address \$C000-\$FFFF.

3: 0=Normal screen, 1=Shadow screen.

4: ROM select: 0=128K Basic, 1=48K System.

5: If set, further output will be ignored until Reset.

6-7: Unused. Should always be set to 0.

The running emulation will honour Bit 5, but this command is effective regardless of the state of Bit 5.

1.137 ARexx command: GETPAGE

Syntax: GETPAGE

Arguments: None:

Result: Value last output to port \$7FFD

Return code: OK

Function: Gets the value last output to port \$7FFD which controls 128K paging features. The bits of this port have the following meaning:

Bits: 0-2: RAM page number to map at address \$C000-\$FFFF.

3: 0=Normal screen, 1=Shadow screen.

4: ROM select: 0=128K Basic, 1=48K System.

5: If set, further output will be ignored until Reset.

6-7: Unused.

1.138 Demo script: LoadSCR.asprx

This is a very simple script which takes a .SCR file and writes the data into the display RAM region of the emulated Spectrum. As soon as you RUN the emulation you will see the Loaded .SCR image, but it will be lost as soon as the running program writes to the screen. ie: You should only use this when the Spectrum display will remain static.

1.139 Demo script: SaveSCR.asprx

This demo script reads the RAM contents of the current Spectrum display and saves the data to a .SCR file selected by the user. Since a .SCR file is a simple RAM dump of the display region, this is a very simple procedure!

1.140 Demo script: POKEr.asprx

This demo script allows you to enter POKE values at any time.

Running the script will open a small window in which you can type a comma-separated address and byte value to POKE into RAM. Enter a blank line to Quit.

This is a simple basis of a script to accept multiple POKES as game cheats etc.

1.141 Copyrights and Distribution

ASp is programmed by and copyright Ian Greenway.

The Spectrum ROM images are copyright Amstrad Plc.

MuLib and MuTools are copyright Thor Software.

The XFD package is copyright Georg Hörmann & Dirk Stöcker.

AHI is copyright Martin Blom.

All other copyrights acknowledged.

Important information concerning the copyright status of Spectrum software.

Distribution:

The archive may be freely distributed in unmodified form. No files may be added, removed or modified, with one exception: The Spectrum ROM files may be removed, but may NOT be replaced with modified or alternative files.

ASp is Freeware. It is free to use, but the author retains copyright. It may not be distributed in a form which requires payment beyond typical media costs, nor can it be used to promote commercial software.

Disclaimer:

All parts of the ASp package are provided without warranty of any kind. The entire responsibility for use of this package is with the user. No responsibility is accepted by the author or any other individual or group for any consequence of the use or performance of the package. No representation is made for fitness for purpose, reliability, accuracy or any other aspect of performance. If these terms are in contravention of any local laws or are otherwise unacceptable, do not use this package.

In other words, ASp is provided in the spirit of "Copyrighted Freeware". All features are provided in good faith, but at the end of the day it is free and you get what you pay for. If it makes your harddrive blow up or starts World War Three, don't go looking at me! :)

1.142 Credits

A lot of people have been very helpful during the creation of this program.

Special thanks, for their huge amount of help and support, go to:

Stephen "FBlit" Brookes

Thomas "Thor" Richter

Thanks to all the others who have contributed at one stage or another:

Martin Blom

Mac Buster

David Cripps

Andrew Deacon

Tomas Hauerland

Martijn "void" van der Heide

BKJ/Vulture

Tomaz Kac

Luca "Hexaae" Longone

Mario Sarno

Jools "Buzz" Smyth

Gerard Sweeney

Neil Whitehead

Everyone on the Spectrum Emulator mailing list.

...and any other ppl I might have forgotten to mention.. ;-)

A special thank-you also to everyone who has contacted me to say they use ASp.

Information from:

A Z80 Workshop Manual - E.A. Parr

MC68030 User's Manual - Motorola

The "TechInfo.doc" from the "Z80" emulator - Gerton Lunter.

The comp.sys.sinclair FAQ - The Cabal.

Various anonymous documents on the internet.

And my trusty rubber-keyed Spectrum 48K, of course!

ASp NewIcon images by Lee Cook.

ROM images (c) Amstrad Plc.

ASp has been written from scratch in 680x0 asm using HiSoft's Devpac 3.18, on a 32Mb, 64MHz 68060, A1200.

Some of the data structures have been created in TurboCalc2 and processed using programs compiled with VBCC.

Total size of all source and data files used to create ASp: 1.35Mb.

Size of asm source files for final assembly: 774Kb.

Approximate assembly time in Devpac: 7.5 sec.

1.143 Version History

Most recent changes first.

V0.84: Add: AHI 16-bit sound.

Add: Screen-grabber.

Chg: Runs in 48K mode without MuLib.

Add: gameport.device joystick access.

Fix: DAA instruction now exactly correct.

Chg: Major rewrite of graphics system.

Chg: EmuCore now has no secondary decodes.

Fix: ROM manager failed to release empty slots.

Chg: TZX loader res. now 64T. (Fix Simpsons Vs. Space Mutants)

Chg: Prefs now tooltype-alike text file.

Chg: Removed almost all CLI Args.

Fix: JoyKeys display was bad if no initial settings.

Chg: Screen/palette fix for some truecolour displays.

Add: Ctrl key is E-shift.

Fix: Z80 loader with lone \$ED at end of compressed block.

Fix: SNA loader would not always correctly clear stack.

Fix: Colour-0 border would leave old colour if turned off.

Add: Option to swap stereo channels.

Fix: Flash-loader would lock out Spectrum if no tape.

Chg: Some changes to fallbacks and defaults in snapshots.

V0.83: (Abandoned)

New concept for graphics updates and higher resolution rainbow.

V0.82b: Fix: 128K memory port mapping.
Fix: Harmless overrun in InitZXKeyMatrix().
Fix: Issue 2 may not have worked.
Chg: Redo some emuroutines for speed.
Chg: Revert to lowlevel.lib installed interrupt.
V0.82: Add: ROM Config. GUI.
Add: Selectively write-enabled ROMs.
Add: SP file load and save.
Add: Decrunches ROM files on load.
Add: Dragndrop for Tape files.
Chg: Allow corrupted TAPs to be loaded.
Add: Simple status bar.
Add: TZX Archive Info block.
Add: TZX User Select block.
Add: TZX Instructions block.
Add: TZX Picture block.
Add: User choice of gameport access method.
Chg: Lowlevel.library now optional.
Chg: New algorithm for some Z80 8-bit math.
Add: 48K Issue 2 support. (Port \$FE bit dependency.)
Chg: Use MAPP_SHARED mode of MuLib for FScreen.
Chg: Rewrite flashloader from scratch. (Fixes Lemmings.)
Fix: TAP2TZX could truncate data.
Chg: Speeded up initial TZX block scan.
Fix: No longer holds locks on files in use.
Add: Tape browser is now AppWindow.
Add: Share menus with tape browser.
Add: Navigate tape with keys.
Add: EXTTEXTVIEWER and EXTIMAGEVIEWER tooltypes.
Add: ROMPAGE tooltypes, etc.
Add: Ability to have alternate names for tooltypes.
Add: Secondary ROM trap for Flashloader. Avoid white border.
Chg: Clean up and modify save format selection.
Chg: ARexx SAVE uses new system too.
Fix: Port \$7FFD lock bit was ignored in 0.81.
Fix: TZX Custom title now full 16 chrs.
Chg: Internal reworking of file and string handling code.
Fix: Various corrections to docs.
V0.81: Add: Simple tape browser.

Fix: LDxR can now self-modify on the fly. (Alcatraz loader.)

Fix: T-state accum. error in FDED/DDED "unimplemented" inst's.

Fix: Z80 loader now stays in current 128K mode if possible.

V0.80Alpha:

Add: TZx loader support.

Fix: Corrected errors in R-register count for block inst's.

Chg: Rewritten Z80 I/O routines for consistency.

V0.79b: Fix: Enforcer hit on startup with no ROMPATH tooltype.

Fix: Internal data alignment problem.

V0.79: Chg: Total re-write of ToolType, CLIArg and Prefs handling.

Chg: Mods to init. code to suit new ToolTypes.

Add: Ability to load snapshot from a project icon.

Add: Ability to load snapshot from a CLIArg.

Add: Drag-and-Drop (AppWindow) snapshot loading.

Add: Redo JoyKeys to work for CD32 buttons.

Add: Logarithmic AY volume option.

Fix: Current Dir confusion in some cases.

Fix: 48K-mode SNA save hasn't worked since V0.75!

Fix: Potential buffer overrun in program name.

Fix: JoyKey settings were not always saved in prefs.

Fix: INxR instructions were broken.

Chg: Manually optimised 0(An) where Devpac doesn't.

Add: Overwrite? warning on Save.

Add: Workaround for possible P96 MoveWindow() bug.

Chg: Minor modification to DAA instruction H-flag.

V0.78: Add: Load and Save of settings.

Add: Load snapshots via XFD lib. (optional)

Add: Speed synch selection.

Chg: Minor internal changes.

V0.77: Add: Window Mode. (RTG "Window on Workbench")

Add: Top and bottom borders.

Add: Centre screen options.

Add: Colour 0 border re-introduced.

Add: Adaptive rendering and benchmarker.

Add: Larger ZX->Chunky lookup table.

Add: Anti-aliasing to audio.

Add: "No Audio" feature, using timer.device for synch.

Chg: Remove contention from port \$FE.

Chg: Minor modifications to Hi Accy Timing code.

Fix: Crash on startup when no icon present.

V0.76: Add: Rewrite display rendering to support RTG.

Add: Beam synchronised graphics update.

Add: 128K+2 mode.

Add: CLI Args.

Add: More ARexx commands, including those which return values.

Add: Ability to run ARexx scripts.

Add: Definable ARexx port name.

Add: Higher accuracy timing.

Add: "Smart Kempston" stick option.

Add: More flexible ROM loader.

Add: Audio Filter control.

Add: ZipWindow gadget.

Add: Very simple Audio-Save.

Fix: NOFKEYS option now OFF when no icon.

Fix: Minor bug in Load routines.

Fix: Sinclair 2 Joystick now correct.

Fix: Loading without STOP/RUN now uses correct CPU timing.

Fix: Z80 Load/Save routines with uncompressed data.

Fix: Memory leak in GUI code.

Fix: Window refresh was broken.

Fix: EI no longer allows INT directly afterwards.

Chg: Make ARexx port name visible.

Chg: Rearrange emu control to avoid potential (rare) load problem.

Chg: 48K mode now uses "real" 48K ROM, not 128K ROM#1

Chg: Small mods to GUI font-sensitive scaling code.

V0.75b: Fixed MCP Enforcer hits with BLANKPOINTER. (Luca Longone)

Fixed lockouts in the GUI.

Fixed menu command keys for Rewind, Reload and Run.

Fixed T-times for single-shifted opcodes which broke in V0.75. :(

Fixed potential to lose an audio event after the end of a frame.

Added "Keep Kempston interface" menu item.

Move Reset from F10 to F9. (Lee Cook)

Add "NOFKEYS" tooltip.

Add a LendMenus() to make it easier to access the menus when dragging down the system screen.

Add correct T usage for successive DD or FD opcodes.

Add interrupt is only valid for 32T.

V0.75: Change Z80 core to obtain 15-20% speed increase on 68030.

Added simple ARexx port.

Add more joystick types.

Add user-defined "joy-keys" window.

Change Kempston decode for compatibility. (Artur Chlebek)

Add BLANKPOINTER tooltype.

Fix multiple menu selections. (Luca Longone)

Added variable audio sample rates.

Added timing changes for 128K/48K modes.

Add Quiet 48K option, default changed to Loud.

Fix block IN and OUT flags and timings. Fixes Gryzor music.

Corrected timings for ZX interrupts.

Fixed loading of Z80 files of 49179 bytes.

Optimised AY sound build again.

Changed version string to correct format. (Luca Longone)

V0.74: Much faster 128K bank switching.

Add support for V42 mmu.library.

Re-introduce 68030 speedups based on ASp V0.71b.

Z80 loader will load the initial part of a SLT.

F2 (ReLoad) changed to F6. (Grr!)

Better error report on init failure.

V0.73dβ: Fix: Menus again! Forgot a MENUTOGGLE flag.

V0.73cβ: Internal development version.

V0.73bβ: Fix: Fixed NewLook menus rendering as black-on-black.

Fix: If AGA wasn't available, the OCS display was broken!

V0.73B: Fix: 128K paging.

Fix: Shadow screen swapping.

Fix: Glaring major bug in TAP loader.

Fix: Z80 loader bugs.

Add: Z80 saver.

Add: Auto-detect and option for SNA or Z80 save.

Add: 48K-128K mode switch.

Add: Menus.

Add: Stereo sound.

Add: About function.

Enhance GUI appearance.

Speed up sound building.

V0.72B: Prerelease 128K version.

Add: 128K paging.

Add: 128K file loading.

Add: Lots of bugs. ;^)

V0.71b: Experimental speedup. 5-15% depending.

V0.71: Fix: Rewrote sound handling. No longer uses audio.device calls.

Added ToolTypes: LOADPATH, SAVEPATH, ROMFILE.

Added Reload and Rewind functions.

Key shortcuts for Load, Save, Reload, Reset.

System screen now *activates* input!

Fixed "." on numeric pad.

Test for AGA chipset.

Filerequesters are a bit nicer.

Inactive GUI buttons are ghosted.

Some potential deadlocks removed.

V0.70: Speeded up sound routines.

Now all audio goes through audio.device! 100% system friendly.

Allows running multiple copies at once. :)

Add emulation of INI/OUTI/IND/OUTD.

Generalised i/o port decoding.

V0.69: Internal Test Version. Added AY-3-8912 sound emulation.

V0.68: Beta Release only. Limited ".TAP" loading support.

V0.67: Now has choice of Custom AGA Display or Native System Screen.

GUI: Added STOP button and Display-Type cycle gadget.

Can now Load/Save/Reset even if emulator is running.

Now requires WB3.0 (V39) or above.

Lee Cook contributed an alternative NewIcon. Thanks!

V0.65-V0.66: Internal.

V0.64: Interim bugfix release:

Fixed a nasty enforcer hit in the .SNA loader! Oops! :-)

Added facility to load ".Z80" files, versions 1, 2 & 3, 48K type only.

V0.63: Added some error requestors.

Fiddled around with parts of the source to aid future updates.

V0.62B: Made emulation code more regular at minor cost in speed on '030.

First public release.

1.144 Frequently Asked Questions

Here are some issues that could or do get raised in relation to ASp. They are not in any particular order, so you'll have to scan through...

Q: Will you be making an OS4-native PPC version of ASp?

A: Not for a while, if at all. ASp is coded in 680x0 assembly language and so would require a complete rewrite to support PPC code. It is possible

that OS4-specific features may be introduced where thought beneficial, but it is to be hoped that ASp will remain compatible with OS3.x.

Q: Why doesn't ASp run in 128K modes on OS4/MOS/UAE?

A: 128K Spectrums can shunt whole "banks" of RAM into and out of the 64K address space of the Z80 CPU. In order to obtain a good emulation speed, ASp uses the 680x0 MMU, via `mmu.library`, to simulate this behaviour. At present, "next-gen" Amigas and emulations do not support the MMU.

It is possible to have this functionality without the use of an MMU, but with a speed penalty. Since the target systems are likely to have more than adequate CPU power, it is hoped that ASp will support alternative schemes soon.

Q: ASp is a bit glitchy and jerky when using UAE?

A1: It is strongly recommended to use the UAE-specific AHI and P96 drivers supplied with the UAE package. ASp has been seen to run comfortably using a modern PC with these installed.

A2: There are interactions between the UAE display refresh and ASp's display refresh that can make for untidy 50fps "smooth" Spectrum graphics. If ASp is used in full-screen mode, the open/closing of the screen can be slow.

These are consequences of using an emulated system.

Similarly, there can be inexplicable pauses from time to time, which are probably more attributable to Windows than UAE.

Q: When using ASp, I get stuck keys / mad mouse movements / random window scrolling / blood oozing from keyboard / etc.

A: Typically an incompatibility with `lowlevel.library` on your system. Try using the System Pad or System Stick (as appropriate) access schemes, and failing that, use Gameport or Direct Stick mode. Remove `lowlevel.library` and reboot if required. If it still screws up, let me know.

Q: When switching from `lowlevel.library` to `gameport.device` joystick access modes, ASp refuses to run the emulation.

A: `lowlevel.library` opens `gameport.device` also. Some versions never close it again, meaning no other program can use the `gameport.device` until a reboot. I'd regard that as a bug in `lowlevel.library`, but there are suggestions that it is deliberate.

Q: I have a TAP file that ASp insists is corrupted. Other emulators accept it.

A: ASp will notify you of damaged TAP files, but will give you the option of loading them anyway. Whether they then work properly is another matter.

Q: I have a TAP file which ASp accepts, but it does not load into the Spectrum properly.

A1: Several TAP files have been generated in conjunction with a hacked version of a program. The hacked code tries to load "standard" (ie: .TAP) data instead of the correct protected loader (.TZX) data. ASp should deal with this in general, but many such files have been found to be totally corrupted. Check the code works with other emulators, or better still, get the .TZX version.

A2: Some older programs which can be correctly represented using a .TAP file do not like running with later model Spectrums. Unlike a snapshot, a tape file does not contain instructions to tell the emulator what machine type to use. Try running the most basic Spectrum hardware config. possible.

Q: I have a TZX file which does not work properly with ASp.

A: ASp's TZX support is incomplete. Please check the section detailing the extent of ASp's compatibility. Whether or not ASp should deal with it at the moment, it would be helpful to know where the incompatible file can be found to aid with future development.

Q: Will you support TR-DOS or other disk system?

A: Eventually, yes, but a disk system is a lot of work and is of limited interest to me personally. The standard +3DOS is likely to happen before TR-DOS, which is sparsely documented.

Q: I get a "tear" or glitch in the moving graphics. I thought this emulation was supposed to be beam-synched?

A: The emulated Spectrum video beam is synchronous to the emulated CPU. However, the display update does not happen synchronous to the Amiga video beam. What you are seeing is the update rendering happening as the video beam passes that point.

Q: The screen update seems jerky. I know this software on a real Spectrum scrolls the display perfectly smoothly. What gives?

A1: You are using an Amiga ScreenMode with a vertical refresh other than 50Hz. The Spectrum only knows about 50Hz frame speeds and so only updates its display in time with that. The interaction between the two frame rates causes some frames to be visible once, others twice, some not at all. This results in a jerky display.

A2: If you have insufficient CPU power available to ASp, or you are running other programs which periodically steal the CPU at the same time, ASp cannot maintain full Spectrum speed. Since it must finish emulating one frame before starting on the next, the frame update rate is inconsistent and slow.

A3: If you are using FScreen or some other periodic display refresh system it will limit the maximum display update frequency. In the case of FScreen, try using the "Force Chip Screen" option to bypass it.

Q: The rainbow effects in program XYZ don't look like they should.

A1: Make sure the program was designed to run on the machine being emulated. The timings of 48K and 128K machines are different, and may be different to other clones, like Scorpion or Pentagon. Some "128K" demos written for the clones will actually run in 48K mode with AY enabled, and will have better rainbow effects.

A2: ASp's rainbow effects are approximated. Taking account of every possible timing issue and calculating every display pixel based on it is not practical to do without using huge amounts of CPU time. The system used in ASp tries to work in most cases with minimal additional CPU load.

Q: The graphic detail and/or colour in Window mode seems wrong compared to what I see in Screen mode.

A1: ASp is not always able to allocate accurate enough pens to display a good image in Window mode. Try increasing the depth of your Workbench screen (or whatever screen you have the ASp window on), or freeing up some sharable pens.

A2: If your screen mode is interlaced or your flickerfixer is active for this mode, you can sometimes see interactions between the Spectrum display updates and the Amiga display updates when 50 frames/sec Spectrum graphics are being displayed. This can show up as missing, flickering or torn graphics, or even wrong colours.

Q: I have a game which is supposed to work with Kempston Joystick but it just goes nuts when I run it!

A1: Try the Smart Kempston stick. It may help in some cases.

A2: Some games have an as-yet unidentified problem with most emulators' Kempston implementations. Use an alternative control method if possible.

Q: I have written an ARexx script to control ASp, but it always complains that ASp's "Host Environment not found".

A1: By default, ASp names its ARexx port with a suffix, 0-9, to allow multiple copies to run simultaneously. Alternatively it is possible to explicitly specify a name. Make sure your script knows which port name ASp is using.

A2: Calling the script from within ASp (Project->Run ARexx...) will always be able to address ASp's port by default, regardless of its name.

Q: XYZ program crashes, misbehaves, or otherwise goes wrong.

A1: Are you sure the snapshot you are using is not corrupted?

A2: ASp doesn't have a perfect emulation, although it's as accurate as possible. Send me a detailed report of how to reproduce the problem. If you can determine exactly which CPU instruction or flag is causing the problem, so much the better!

Q: I have a .Z80 snapshot file which loads into other emulators but not into ASp.

A: ASp's .Z80 loader is very fussy. Some emulators will save out files in an incorrect format. This often results in loss of data, although the program appears to work for the most part. ASp currently only loads files which adhere strictly to the file format specification. In future there may be a "tolerant loader" option added.

Q: I have saved a file from ASp which ASp refuses to load back in!

A: Argh! :-) Please contact me with details.

Q: The sound/music in program XYZ is juddery or broken.

A1: You have insufficient CPU power to emulate this program at full speed. Try reducing the sample rate, or increasing the frameskip.

A2: Some audio hardware requires substantial buffering when used with AHL. Try increasing the audio buffer to a larger value.

Q: I can't seem to get ASp to load files from an XFH partition I have set up. The docs say it should work. What's wrong?

A: You need to use XFH V1.35 or better. ASp uses the AmigaDOS 2.0+ function ExamineFH() which requires the underlying filesystem to support the packet "ACTION_EXAMINE_FH".

Q: Will you share the source code for ASp? Will you release it publicly?

A: At the moment, I don't want to give sections of the source to third parties for use with "competing" projects. I feel our already small community would benefit more from a choice of different software rather than a small pool of common software.

Eventually, I think I will release the source publicly or pass it in its entirety to someone who wishes to continue with it.

Q: Why are you spending time coding one old computer to emulate an even older one? Have you nothing better to do?

A1: Some people seem to like it...

A2: No!

1.145 Contended RAM

Spectrum models have a "ULA" chip which concerns itself with several things, including generating the video display. In order for it to be able to do this it has to have priority in accessing the display RAM region(s).

Since the main Z80 CPU can therefore be temporarily blocked from accessing these memory areas they are known as "contended" memory regions.

Running code which resides in this memory or accessing data in it will affect the time taken by the CPU to perform the operation. The exact effect depends on when the access takes place in relation to the video beam.

Since multicolour "rainbow" effects by necessity access the display region throughout the video frame and are time-critical to achieve the desired effect, such timing issues need to be taken into account to reproduce realistic rainbow effects in an emulator.

1.146 Information about ASp's use of mmu.library

The mmu.library by Thomas Richter is an Amiga shared library which allows more than one program to control the MMU without unpleasant interaction.

Historically, the Amiga OS has not offered support for arbitration of access to the MMU hardware. Software wishing to use it has been forced to poke around in proprietary MMU tables, attempting to interpret what is already there in order that it may add its own functions. In a lot of cases this would work when one program did it, but if two or more attempted it, chaos could result.

Using the mmu.library to access the MMU provides two definitive advantages. First, the coding of the MMU is abstracted behind a well-defined API, whereas writing real code for the real thing can be tricky. Secondly, a program accessing mmu.library gets all the benefits of having an MMU without having to concern itself about other running software. Programs using the MMU via the library are guaranteed not to trip over each others MMU code.

ASp uses the MMU to provide the 128K paging feature. In a 128K Spectrum, hardware exists to cause "pages" of RAM to appear in a common address space. In some cases, the same page can appear in two places at once! There are ways to achieve this without the use of an MMU, but they are slow and/or problematic.

If mmu.library is not available, ASp will revert to 48K-only mode.

ASp requires at least Version 43 of the mmu.library.

1.147 What's an MMU?

The MMU is the Memory Management Unit of the CPU. 68030, 68040 and 68060 CPU's have them built in, 68020's can have a separate chip to perform the function. An MMU is not to be confused with an FPU (Floating Point Unit), which is very different. ASp neither requires nor uses an FPU.

The action of an MMU is to take a "logical" (ie: what the running program asks for) address of a RAM region and, transparently, convert that to a physical (ie: real) address in real RAM where the data resides. It is also capable of assigning characteristics like Write-protected to the RAM region.

An MMU is the core of a Virtual Memory system. Data in memory can appear to be in a different place to where it is really stored. Each program can be made to believe that the entire CPU address space is available for its own use, regardless of the fact that only certain portions are "really" available. This is often combined with swap-memory, where data currently unused by the running program(s) gets saved to disk and other data put in its place. The programs that wish to access that data do not know that it has been transferred to disk. When they try to access it, the data gets read from the disk, and may get placed in a different area of RAM, but the MMU quietly converts the requests for the old area into accesses to the new area. The running program is unaware of the change.

ASp exploits these abilities by maintaining an image of the entire 128K Spectrum RAM area and then dynamically mapping selected pages into the address map used by the emulated Spectrum. This allows the CPU emulation core code to remain simple and therefore fast.

1.148 About the XFD package

XFD is an Amiga shared library (xfdmaster.library) and associated tools, originally conceived and written by Georg Hörmann, but latterly maintained by Dirk Stöcker.

By the use of modular decrunchers, either internal to the xfdmaster.library or as external "slaves", the package presents a simple programmer interface to transparently decrunch files in virtually any known Amiga data packer format.

If you have not installed the XFD package ASp will load non-crunched files but will be unable to read crunched files.

ASp requires version 36 or higher of xfdmaster.library.

1.149 About the AHI package

AHI is a retargetable audio system for the Amiga, written by Martin Blom.

Programs that use AHI can output audio to any supported sound hardware the user may have. Users with a basic Amiga configuration will be able to use the native Paula drivers, and users with soundcards can use the potentially superior output from them.

AHI has also been adopted by OS4 and some Amiga emulations, permitting a degree of flexibility and future-proofing in the ASp audio.

ASp will run without AHI installed, but will revert to "No Audio" **channel mode**.

ASp requires version 4 or higher of AHI.

1.150 World Of Spectrum

The World Of Spectrum Website is the official site for all things relating to Spectrums. There are large numbers of documents about everything from game instructions to detailed workings of a Spectrum. There are box scans, adverts from old magazines, cover scans of magazines, famous names of the times and of course, a huge archive of downloadable programs and games for use with the Spectrum.

The maintainers of this site have gone to great lengths to try to get the agreement of the copyright holders for the information contained on it.

You can go there by looking at the URL:

<http://www.worldofspectrum.org>

1.151 ASp Support Web Pages

Updates, extra information, beta versions, prereleases, ROM images, rambling waffle, all are available from time to time from the ASp support pages, part of my humble website (note new address than previous versions).

Take a look at:

<http://web.ukonline.co.uk/iang/>

1.152 Contacting the Author

ASp is written by Ian Greenway.

You can contact me by email at: [iang @ ukonline.co.uk](mailto:iang@ukonline.co.uk)

Feel free to offer comments, suggestions, support, criticism, etc. I cannot guarantee your suggestions will ever be implemented, but all suggestions are considered.

I am also happy to answer any problems or questions you have with using ASp. Please read the appropriate guide section and the FAQ's before asking, though, that's why I spent hours writing them! :)

Oh, and if you've got any money, women, fast cars, etc. you don't want, I'll take 'em off your hands! ;-))

Also, check out [my website](#) .

1.153 Index

#

48KISSUE2 argument

A

access method of gameport

adaptive frameskip

ADAPTIVEFS argument

add AY chip to 48K model , menu item

ADD48KAY argument

ARexx

commands

portname, defining

scripts

AREXEXPORTNAME argument

ASp web pages

audio

channel mode

options

save mode

AUDIOMODE argument

AYCHAR argument

author, contact

B

BLANKPOINTER argument

Browse... , GUI function

Browser, tape

C

calibration of graphics rendering

CENTREX argument

CENTREY argument

centring , screen

channel mode , audio

COL0BORD argument

colour zero border option

command line arguments

contended RAM

copyrights

credits

[cursor joystick](#)

[custom keys joystick](#)

D

[delay during flash-load](#) , menu item

[disclaimer](#)

[DISPLAY](#) argument

[display type](#) selection

[distribution conditions](#)

E

[Eject](#) , GUI function

[Enable F-Keys?](#) menu item

[EXTIMAGEVIEWER](#) argument

[EXTTEXTVIEWER](#) argument

F

[FAQ's](#)

[features](#) , overview

[flash-load standard blocks](#) , menu item

[FLASHLOAD](#) argument

[Force chipmem screen?](#) , menu item

[FRAMESKIP](#) argument

[frameskip setting](#)

G

[GETPAGE](#) , ARexx command

[GETVERSION](#) , ARexx command

H

[HELP](#) , ARexx command

[history, version](#)

I

[icon tooltypes](#)

[INSERT](#) , ARexx command

[Insert...](#) , GUI function

[installation](#)

[introduction](#)

[issue 2 \(48K\) support](#)

J

[JOYPORTACCESS](#) argument

[JOYSTICK](#) argument

[joystick hardware](#) access method

[joysticks](#)

cursor

custom keys

Kempston (normal)

Kempston (smart)

Sinclair 1

Sinclair 2

K

keep Kempston interface , menu item

KEEPKEMPSTON argument

Kempston (normal) joystick

Kempston (smart) joystick

keyboard, usage

L

LOAD , ARexx command

Load... , GUI function

Load snapshot prefs? menu item

loading snapshots

LOADPATH argument

M

MACHINE argument

machine type selection

Make 48K Issue 2? , menu item

MMU , description

mmu.library

More... info button

N

O

P

PEEK , ARexx command

POKE , ARexx command

Preferences , loading and saving

Play/Pause button

Q

quiet 48K beeps , menu item

QUIET48 argument

QUIT , ARexx command

Quit , GUI function

R

READRAM , ARexx command

RELOAD , ARexx command
ReLoad , GUI function
remember position option
requirements
RESET , ARexx command
Reset , GUI function
reset on machine change , menu item
REWIND , ARexx command
ROM Configuration GUI
ROM images required
ROMPAGE0 argument
ROMPAGE1 argument
ROMPAGE2 argument
ROMPAGE3 argument
ROMPAGE4 argument
ROMPATH argument
RUN , ARexx command
RUN , GUI function
S
SAVE , ARexx command
Save... , GUI function
saving snapshots
SAVEPATH argument
screen centring options
SCREENDDEPTH argument
SCREENMODE argument
screenmode selection
Settings load and save
Sinclair 1 joystick
Sinclair 2 joystick
SP , snapshot format
Spectrum, description
SPEED argument
Speed adjustment of emulation
SNA , snapshot format
snap to grid? option
snapshots
legalities
loading

[saving](#)

[Start-up](#) behaviour

[STOP](#) , ARexx command

[STOP](#) , GUI function

[SWAPSTEREO](#) argument

[Swap stereo channels?](#) , menu item

T

[TAP](#) files

[using](#)

[file format](#)

[TAPDELAY](#) argument

[tape browser](#)

[tooltypes](#)

[TZX](#) files

[using](#)

[file format](#)

[supported blocks](#)

U

[usage guide](#)

[USEFKEYS](#) argument

V

[version history](#)

[video configuration options](#)

W

[Window display mode](#)

[World Of Spectrum website](#)

[WRITERAM](#) , ARexx command

[Website](#) , ASp homepage

X

[XFD](#) package

Y

Z

[Z80](#) , snapshot format
