

RamUtils V0.95 By P.Harvey-Smith.

Purpose.

To provide a largely compatible drop in replacement for Acorn's sideways ram utilities as included with 1770 DFS. With the added advantage to not using main ram, so PAGE remains unchanged.

Also to provide extra related functionality.

Commands and syntax

Note all numbers provided to the commands must be specified in hexadecimal. Since these commands are meant to be generic, and not depend on a particular implementation of sideways ram, they do not provide pseudo banks W,X,Y,Z as some implementations of the Acorn commands do.

***ROMS**

Syntax:

*ROMS
*ROMS <id>
*ROMS <firstid> <lastid>

Purpose:

To display a list of all banks of sideways memory containing recognised rom code.

Description:

*ROMS with no parameters will list all recognised rom banks from 0 to F. Supplying a single rom id will list the information for just that bank, supplying 2 rom ids, will list the information for all roms found between the two ids.

***RESET**

Syntax:

***RESET**

Purpose:

To reset the machine as if by CTRL-BREAK.

Description:

*RESET will reset the machine, this is useful in two situations, the first is a convenient way of resetting an emulated bbc. The second is being able to reset programmatically from a script file, or from a basic program.

***SRCOPY**

Syntax:

*SRCOPY <id> <id>

Purpose:

To copy the contents of one sideways bank to another.

Description:

SRCOPY allows you to copy the contents of one bank of sideways memory to another, obviously the destination bank must be ram, but the source bank can be either rom or ram.

***SRCMP**

Syntax:

*SRCMP <id> <id> (<start addr>) (<end addr>)

Purpose:

Compare to sideways banks for equivalence

Description:

Allows two banks of sideways memory to be compared for equivalence, on exit the first non matching address will be printed, this will be in the &8000-&BFFF region if there is a mismatch or will be &C000 if the banks are equal. If the optional start and end address are supplied then only this region will be checked, otherwise the default &8000-&BFFF

***SRLOAD**

Syntax:

*SRLOAD <filename> <address> <id> (<Q|S>) (<I>)

Purpose:

Load a file into sideways ram.

Description:

*SRLOAD allows the loading of a disk file into a bank of sideways ram, please note that the address specified **must** be in the &8000 to &BFFF area of memory. The loaded file can be any type of file, but it should be noted that if you are loading an image of a rom this way, you will need to reset the machine (either with CTRL-BREAK or *RESET) before it will be active.

The following optional flags may be specified :

- | | |
|---|---|
| Q | Use the ram in the OSHWM to HIMEM region as buffer instead of default. |
| S | Use the screen memory HIMEM-&7FFF as buffer |
| I | Initialise the rom after loading, this may allow some roms to work immediately and not need a CTRL-BREAK to initialise. |

Both Q and S can lead to dramatic improvements in loading speed, as if enough memory is available they will use the OSFILE MOS call to load the file which is about 5 times quicker than the default which uses OSGBP.B.

***SRREAD**

Syntax:

*SRREAD <dest start> <dest end> <sram start> <id>

*SRREAD <dest start> <+dest length> <sram start> <id>

Purpose:

Transfer a block of sideways memory to main memory.

Description:

*SRREAD allows a block of sideways memory to be copied to main memory, note you should have reserved the main memory that you are copying to avoid over-writing vital system information ! Note also that the destination and sram addresses are range checked, the destination start and end must be in the &0000-&7FFF area, additionally if the destination end is specified as a length then the resulting end address must also be in the same range. The sram address must be in the &8000-&BFFF area.

***SRSAVE**

Syntax:

*SRSAVE <filename> <start addr> <end addr> <id> (<Q|S>)

*SRSAVE <filename> <start addr> <+length> <id> (<Q|S>)

Purpose:

Save a block of sideways memory to a file.

Description:

*SRSAVE can be used to save a block of sideways memory to a file, the specified start and end addresses (or start + length), must be in the &8000-&BFFF area. Note that this can be used to save from any bank, irrespective of if it is rom or ram.

The following optional flags may be specified :

- | | |
|---|--|
| Q | Use the ram in the OSHWM to HIMEM region as buffer instead of default. |
| S | Use the screen memory HIMEM-&7FFF as buffer |

Both Q and S can lead to dramatic improvements in saving speed, as if enough memory is available they will use the OSFILE MOS call to load the file which is about 5 times quicker than the default which uses OSGBPB.

***SRWRITE**

Syntax:

*SRWRITE <source start> <source end> <sram start> <id>

*SRWRITE <source start> <+length> <sram start> <id>

Purpose:

To copy a block of memory from main memory to sideways ram.

Description:

*SRWRITE allows the copying of a block of main memory to sideways memory, the destination block of sideways memory will of course need to be ram, for this to have a useful effect. Note that the source start and end or source start and length must both result in addresses in the &0000-&7FFF and that the sram address must also be in the &8000-&BFFF areas.

***SRZAP**

Syntax:

*SRZAP <id>

*SRZAP <first id> <last id>

Purpose:

To blank an area of sideways ram.

Description:

*SRZAP allows a block of sideways ram to be blanked, this will write &00 to every address in the specified bank, this is useful to remove a previously loaded rom image from sideways ram without having to power the machine off. If two banks are specified all ram banks between them will also be blanked.

It should also be noted that if the RamUtils are loaded into sideways ram, that trying to

*SRZAP that bank will silently fail as it would otherwise crash the machine.

*SRZAP also sets a bank's rom type byte to 0 so it can be used to temporally remove a rom from the chain of roms called by the MOS.

