007
MULTISPY

A unique selection of ELEVEN programs plus a powerful MULTICHOP program, which can genuinely transfer majority of tapes:

TAPE-TO-TAPE

TAPE-TO-OPUS DISC

TAPE-TO-MICRODRIVE

TAPE-TO-SPEC+3 DISC

The versatily of the programs means that by simply changing the LOAD/SAVE command in three programs, tape can be used to transfer to future devices.

Another genuine User Friendly product from

G.A.BOBKER of ZX-GUARANTEED

The 007SPY program also has TWO extra options not mentioned in the Instructions as they are unlikely to be of use. These options are for chopping START or END of Files, but these are ALWAYS Saved out Headerless....Therefore more useful to use the MULTICHOP instead as it ALWAYS Saves out by with Headers on.

KEY H....Used to chop-off number of Bytes from START of a File. Method of use is to LOAD File as usual, then press Break. Press H then enter the number of Bytes to chopped off beginning of File. Press Enter and shortened File saved out. If you Loaded a File which included a screen picture, and used Option H to chop off 6912 Bytes, then File would Save without the screen part of the program.

KEY P....Used to chop off from END of a File. This is different to the above, as you press P then enter number of Bytes to be chopped off the end, THEN play in the File. When Loaded in, press Break and then press S to Save out and it'll save out the File with the end Bytes chopped off.

007 MULTISPY

INDEX

Page Order

Intro	1
007SPY	2,3,4
DE-PULSER	5
DEPULSER 2	6
ditto SPD1/2/3	7
Trans Intro	8
prog1/2/3	9 (28)
BASIC Trans	10
UnMERGEables	11
CODE Trans	11
MEMORY	12
CODE details	13
Example Trans	13
Single CODE	17
HEADERLESS	18
ditto Example	19
Long Progs	20
ditto Example	20
48K Programs	22
Awkward progs	23
Hyper Loads	23
Code Loaders	24
Double MOVER	24
MICRODRIVE NOTE	26
007VAL	27
007RECLAIM	27
prog1/2/3	28
007SEARCH	29
007MOVER	30
MULTICHOP	31

Utilities

007VAL	27
007RECLAIM	27
007FILE	28
prog1/2/3	28
007SEARCH	29
007MOVER	30
MULTICHOP	31

PROGRAM TRANSFERS

CHUKIE-EGG	36
CYRUS CHESS	19
HORACE (SKI)	24
JETPAC	13
KNIGHTLORE	35
RAMBO	33
SCRABBLE	20
SHERLOCK	37
TLL	35
UNDERWURLDE	34
VALHALLA	37
VU-FILE old	38
VU-FILE new	39

SPECIAL ITEMS

MEMORY MAP	12
MICRODRIVE NOTES	26

G.A.BOBKER.
29 Chadderton Drive, Unsworth, Bury, Lancs.

The complete suite of ELEVEN programs on the tape 007 MULTISPY are explained in this booklet. It is a complete reference manual on how to transfer to a drive. This means it is a lot to read, but the reward at the end is an understanding of how to transfer programs not mentioned in this booklet.

Before starting it is useful to have some knowledge of the structure of programs on a tape.

Normal Basic or Code programs have TWO parts:

HEADER — This is 17 Bytes long and tells Spectrum the Name, Length & Program Type. Also the Start location if Code, or Auto Start line if a Basic which auto-starts.

MAIN BLOCK — This is the actual program.

The gap between a Header and Main part is 1 second, The Spectrum knows if it is the Header or a Main Block by the first Byte it reads off each Block. If a 0, it is a Header. If a 255, then its a Main Block. Headerless Files are simply programs WITHOUT a Header and should have a "Code" of 255. In practise, Headerless programs can be made to have a "code" of any value between 0-255. Older programs had all been "normal" and can be copied by using a "Header Reader" to find out the lengths, etc, of the various Blocks. Each part was simply Loaded into the Computer, then Saved out. Even the longest programs had been limited to 16K to ensure they Loaded into the older Spectrum. Then along came the 48K machine. If the program was, say 9000 Bytes long and started at 16384, we could Load it higher by; LOAD "" CODE 30000 and Save to a tape by; SAVE "name" CODE 30000,9000. To Load and RUN the copy, simply reload by giving the original address, I.E. LOAD "" CODE 16384. Even now a huge number of programs can be copied, or transferred to OPUS or MICRODRIVE (or other Disc systems) using this "relocating" method.

The Spectrum CANNOT Load programs which are Header -less or had been Saved at Higher speeds! Games using such will first Load in a Code block which is really a modified copy of the Sinclair ROM Loader routine. This is then activated, and loads in the "funny" blocks. Due to there being thousands of different ways to make awkward programs (such as Speedlok), it is NOT possible to make a tape capable of copying or transferring all games. (The ONLY way to transfer or Copy all is to use the INTERFACE 007 unit). The 007 MULTISPY altho' powerful cannot copy/transfer everything, but CAN copy more programs than any other available. This is why this booklet also includes techniques for various awkward programs which hopefully you can apply to similar games,etc

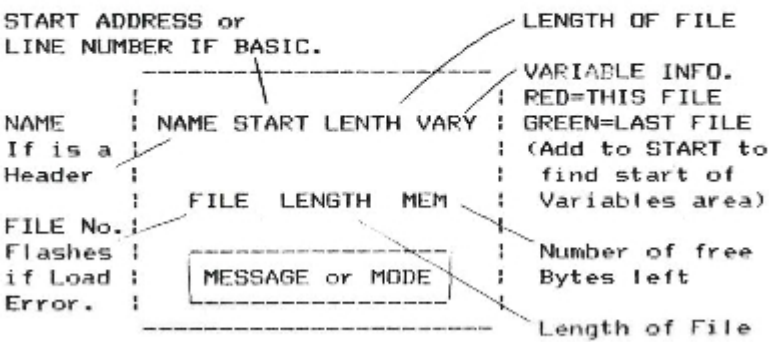
---

GETTING STARTED

Consider first the 007 MULTISPY for use as a tape Copier. Quite a few actions will be applicable later. Start off with a short NORMAL type program first such as JETPAC or HORACE GOES SKIING. Then progress to longer ones such as SCRABBLE. Next a long game such ANT ATTACK or TLL, then a 48K game. For making Back-Up copies of your tape there are 3 programs we could use. The are explained next.

007SPY INSTRUCTIONS

Load by LOAD "007 SPY" When Load, screen shows:



For short programs such as JETPAC, HORACE, etc, simply press L to Load and play in complete game. When all Loaded press the Break key. Place Blank tape in recorder, set to record, then press S to SAVE. Note the way the Name of each part is printed (if a Header played in), etc, and note that each "FILE" (Block of noise on the tape) is treated seperately. This means a normal Basic will be TWO Files.

KEY ACTIONS

Key L....LOADs all Files of a program displaying all info. Press BREAK key to exit.

Key S....SAVEs all Files you have Loaded. (Press Break to Exit). Pressing & Holding P key whilst SAVEing will PAUSE after the File Saved out. Use to make gaps longer.

Key A....AUTOSTART removed from a Basic. ONLY works if the Basic Header was File 1. Note: To use, press A then S key.

Key C....CONTINUES Loading without erasing Files already fed in. Use if you want to load a few Files, then Save them out, then Load more and Save out the lot together.

Key K....KONTINUE SAVE. Allows you to continue Saving remaining Files after Break key was pressed after a SAVE and before last File had been Saved out. Useful in you Load in two games. After SAVEing out the first game, press Break. Put another Blank tape in recorder, then press K and second game SAVEd out.

Key D....DELETE last File loaded in. Could repeat pressing of D to Delete more Files.

Key Y....Y-MODE. Use to copy programs with shorter than normal tones at beginning of the Files. Once this key pressed, ALL further Files fed in will be accepted if normal or short tone, until L is pressed to revert to normal. (Only use Y option if necessary as it is very sensitive and can cause problems if noisy tapes).

Key Q....QUIT. Clears and totally resets Spectrum.

NOTE: TOTAL length of a game such as SCRABBLE is too long to load into 007 all in one go. Solution is to load in all except final very long part, press Break key, then with a Blank tape in a recorder set to record, then press S to Save out. Then press L and load in final part of game. When all Loaded in, press Break and again press S to Save to your tape. With even longer programs, copy all except very long last part by using L to Load (then press Break key when just before start of long final part, and copy out using S). Select E or M mode. Play in long final part then copy out.

NOTE: ONLY USE OPTION E or M TO COPY A SINGLE LONG FILE. CANNOT RETURN TO OTHER OPTIONS AFTER E or M.

Key E...EXTENDED-MODE. Extends copying area by using screen as work space. Copies a single File upto 49114 Bytes long.
KEY M...MAXI-MODE. Copies a single File of upto 65279 Bytes. This is a two phase copier:
Phase 1..Reads in File but only calculates length.
Phase 2..Loads in the File calculated. Enter=Save.

Key F...FAST (HYPER). Switches 007SPY to FAST MODE. Calculates Baud Rate whenever F is pressed. This Should be done on a Fast part of the program, and NOT on a screen$ part. Press F a few times, until the most consistent Baud value is found. Then when this value is on screen, press Enter to exit. Copier now set to this speed for this Fast part. ALL keys perform same functions as before, including E & M modes but at this new Buad Speed. If after loading in the Fast part, the N key is pressed BEFORE pressing S, program will be Save out at Normal Speed, but Copy would NOT Load and RUN! (Could then be transferred to Drive, but you'd have to find the Randomize value). Normal Speed is 1500 Bauds. F key may give a different value as your tape machine speed would be different to machine original tape was made on. (This does not matter).
Key N....NORMAL MODE. Switches copier back to the normal Speed.

DE-PULSER

DE-PULSER copies the OLDER type of "Pulsating programs", such as KNIGHTLORE, UNDERWURLDE, DECATH' ALIEN 8, RAID OVER MOSCOW, DAM BUSTERS, GHOSTBUST' BEACH HEAD, etc, etc. Copies have the daft pulses removed and speed dropped down to normal. This ensures that the copy Loads more reliably than the original. Majority of these Pulsing programs consist of one normal Basic part, a short pulsing part, (which sounds like a Header), then 1, 2 or 3 Files. If more than one Basic part, copy first one using 007 SPY. The main part being at a normal speed, means it could be transferred to an Opus/Microdrive, but it would be necessary to use a Dissassembler to find the address to RANDOMIZE to make program RUN. (A few example are given at end of this booklet).

a. Load by LOAD "DE-PULSER" Title and Copyright message will appear when Loaded.
b. Play in the normal part of your game. When Loaded the message CONVERSION COMPLETE will appear. STOP YOUR TAPE immediately.
c. Place a Blank tape in recorder, set to record, then press the ENTER key to Save out.
c. Press ENTER then play in next part of your game. The short pulsing tone (next on tape) is ignored and the longer pulsing part generally builds-up a picture. (Again ENTER=SAVE). All further parts are Loaded in by pressing ENTER and then Saved out by pressing ENTER key.

DEPULSER 2

This program is ONLY for the latest "Pulsing Tone" programs such as "V", "GREEN BERET", "NIGHTMARE-RALLY", "ALIEN HIGHWAY", "WAR", "GALIVAN", etc.

DEPULSER 2 also lets you alter the Speed of the Back-Up copy. Latest Pulsing games are made up of;

ON TAPE	DEPULSER 2 NAMES	COMMENTS
Header	Normal	These two normal program
Main Part 1	File 1	parts have hidden Code in part 2 which is much
Header	Normal	longer and different to
Main Part 2	File 2	the earlier games.
Short Pulser (Part 3)	File 3	These two parts have the "jerky" noises in their
Long Pulser (Part 4)	File 4	leading tones.

USING DEPULSER 2

1. LOAD "DEPULSER 2" when loaded, title appears.
2. Press Enter then play in your game. First normal part will load in and then gives message
   NORMAL FILE ONE LOADED.
3. Press Enter to save out this File 1 to a tape.
4. Press Enter and play in more of your game tape (next normal part). Stop tape when Loaded. It is very important that you DON'T let tape run into next section yet.
5. NORMAL FILE TWO CONVERTED appears, to show that the File is ready for Saving. (Will Save out at NORMAL Speed).
6. Press Enter to Save this out to your Back-Up tape. When Save completed, message appears;
   Press ENTER to load pulsar files
7. Press Enter (screen goes Black), then play in rest of tape.
8. When all loaded, press S key to Save out to your Back-Up tape (Actually saves out two blocks; Screen$ and then the game).
9. SUPRISE...Press the A key and game will start.

To play your Back-Up copy, Load it by LOAD "" and when all loaded, press A key to start the game.

TO CHANGE BAUD RATE OF BACK-UP

1. Reload Depulser 2 by; LOAD "DEPULSER 2"
2. Press Enter and play in Back-Up tape till message appears.
3. Press Enter to Save out to second Back-Up tape.
4. Press Enter then play in more of original Back-Up tape. (Stop the tape when second part has fully loaded).
5. Select Baud Rate (1,2 or 3). Rate 3 may be too Fast for your tape machine to handle...all should work at Baud Rate 2. (Rate 1 is Normal).
6. Press Enter to Save out the File 2 onto your second Back-Up tape.
7. Press Enter (screen does NOT go black), then use key A and S. A=LOAD S=SAVE, for EACH converted File. Will be necessary to rewind tape slightly each time as only short gaps between the Files. It doesn't matter if rewound slightly too much as it waits for a File beginning. (Cannot press A to restart here).

To play this Back-Up copy, Load it by LOAD "" and when all parts have loaded it waits for you to press the A key to RUN the game. (Some copies may give a weird Border initially...press A to RUN)

As the DEPULSER 2 Saves out programs with a known standard format, these are very easily transferred to Drive using the special Transfer programs on 007 MULTISPY, called "prog1", "prog2" and "prog3" These three programs will transfer the Back-Up copies made at NORMAL speed by the DEPULSER 2. See later notes. The ZX-GUARANTEED products are the ONLY ones which can handle these Pulsing types

OPUS/MICRODRIVE TRANSFER

Altho' these both use the same commands to Load or Save, the major difference is that the OPUS uses NONE of the Spectrum memory. The problem is Stopping games so that you can transfer them, and also that some games overwrite the Systems Variables area making then very awkward. Obviously "Speedlok" and Hyper Loaders give even more problems.

With the Microdrive there are extra problem as it uses 58 Bytes for the extra System Variables it needs. This has the effect of moving the start of your Basic program area from the usual 23755, to 23813. The effect of this is that if the Basic contain machine-code in a REM, addresses in the Code and RANDOMIZE USR values could be out by 58. In the game VALHALLA, adding 58 to all the RANDOMIZE USR's solves the problem. Unfortunately most games cannot be "cured" this easily. Also, whenever a Microdrive command is used (this doesn't apply to an OPUS), the Basic area is temporarily moved up by approx 600 to give a "buffer" for it to use. This is reclaimed after the command finished, but the effect of moving up the Basic by 600 Bytes may have caused it to over-write, and thus wipe-out some machine-code. With games which consist of Basic and Code, with the Code being a few thousand Bytes away from the end of the Basic, this second problem wouldn't occur.

There are several ways to transfer programs to the Opus/Microdrive. With the latest Speedlok types this is very easy as we can use the DEPULSER 2 to remove the daft pulses, then we can use the "prog1", "prog2" and "prog3" on 007 MULTISPY to transfer then. We will therefore consider the latest "Pulsing tone" games first.

TRANSFER PROGRAMS

The THREE programs which are after DEPULSER 2, transfer programs which have been "Depulsed" by the DEPULSER 2 (Normal Speed) onto Cartridge/Disc.

1. LOAD "prog1" then play in ALL of the Depulsed game tape.
2. Place a Blank tape in recorder, set to record, then press the Enter key. Saves out TWO Files with a 2 second gap between them. When action is complete, Spectrum will NEW.
3. LOAD "prog2" then enter the name you want for the game.
4. Play in File tape just made by "prog1", but STOP tape when 'Drive motor starts running. When motor stops, play in next File. Will NEW when completed. (Do NOT remove Disc/Cartridge).
5. LOAD "prog3" (This does NOT auto-run).
6. LIST, then change line 10 so that A$="name"; this name must be the same name used in note 3 previously. (If transferring "NIGHTMARE RALLY" or "WAR", then omit step 8).
8. Press RUN and it will Loads a part from Drive, then Searches for the START for the game (may take a few minutes). Will display this as two values which YOU put into line 50 in place of the two Data values at present in line 50.

9. Delete Lines 11,12,13 & 14 then Save to 'drive by; SAVE*"m";1;"name" LINE 5

Games transferred to Drive by these three programs always wait for the pressing of the A key to start. Some games, when reloaded, may give a weird Border effect...ignore this and press A to start.

The "Pulsing" programs are usually considered to be the most difficult to transfer, yet we have transferred them easily. Now we will have a look at various types of programs, starting with the old NORMAL types...some of the latest programs still use this method, then work our way thro' the more difficult ones. The other programs on the 007 MULTISPY will be introduced and used as necessary.

NORMAL PROGRAMS

QUESTION? How do we know if it's a normal program?
ANSWER. Listen to the program as it is being loaded in. If it has a Header then a Block, then a Header then a Block, AND it sounds like it is at normal speed, then most likely it is a simple normal program.
QUESTION? How do I know how long each part is?
ANSWER. Use the Header Reader built into the 007SPY program. Load "007SPY" then press L and play in your program tape & write down info it gives you, (only need the Name, Start address and Code Length).

****
* PLEASE. PLEASE *

* Please do read all these note and don't just *
* jump to the transferring section. *
****

Firstly we will consider transferring BASIC & CODE seperately, then we'll try a few example programs.

TRANSFERRING BASIC

This can be Loaded by MERGE "". If it is a fairly long Basic, then after it has Loaded in it might take upto two minutes before the O.K. message appears. Then can be simply SAVEd to MICRODRIVE or OPUS. It is possible that the Basic might be too long for Microdrive transfer. The answer here is to reduce length of Basic by VALing all numbers. For example: If program has GOSUB 100 change it to be GOSUB VAL "100". Each time VAL is used, it saves three Bytes. On a long Basic, it can save several thousand Bytes. This takes absolutely ages to do. The alternative is to use 007VAL which is near the end of side 1. Simply MERGE in your Basic (or Load and Stop it), then Load 007VAL anywhere in memory, then RANDOMIZE it. Best method to ensure it doesn't get in the way of your Basic is to put it onto the screen. Therefore you would use
LOAD "007VAL" CODE 16384: RANDOMIZE USR 16384

After a short delay, depending on length of the Basic, the usual O.K. message appears. LIST and you'll see all the Basic has been VALed. Note that for technical reasons it doesn't VAL the DEF FN commands (you can VAL then if you want).

If the Basic locks-up when merged in, use the following method, (could try it on DEPULSER Basic part to prove that it works). This method is guaranteed to stop ANY Basic program....some will still be invisible if INK/PAPER characters have been POKEd into the lines.

1. Enter LOAD "" and play in your Basic, but STOP the tape IMMEDIATELY the program name appears.
2. Press Break key. Enter SAVE "x" then Save out to a blank tape, but STOP tape IMMEDIATELY the computer pauses after it has sent out a short burst of tones. Press break then reset computer.
3. Load in the "x" just Saved, by LOAD "" then play in the rest of the original Basic tape, (which is positioned just after its Header as you stopped it after this in action 1 above).

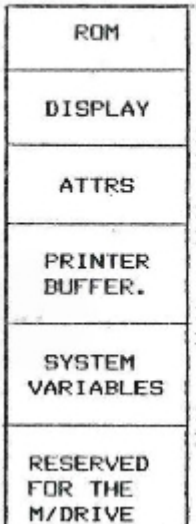
The program will load in but not run. The original auto-start line could be found by Loading original Basic into the 007SPY.

TRANSFERRING CODE

Whenever a Code program is saved you ALWAYS have to include the Start address and the Length. To Save a Code FRED from location 30000 and 123 Bytes long, you would use SAVE "FRED" CODE 30000,123
For this reason we will denote a Code start/length in this manner. A screen$ picture has Code Start of 16384 and Length of 6912 Bytes. In these notes we will refer to this as 16384,6912.

In order to successfully transfer Code blocks we have to know something about the block. This means we have to know the Start address the block Loads into normally, and its Length. Playing the block into the 007SPY will print this info on the screen. We also have to be aware of how the areas in the memory are allocated.

MEMORY MAP



Without an Interface 1 the 48K Spectrum memory map would be as shown.

This area and the screen ATTR (screen colours) are usually lumped together and called SCREEN$. This is 16384,6912

Printer Buffer is 23296,256

These are used by Spectrum to hold info on position of RAMTOP, Print position, etc.

This area expand by 58 Bytes when Interface 1 is on. Also expands when you open a chan for RS232 Printers. Whenever a M/drive using a M/drive command, expands by 596 Bytes

WITHOUT Interface 1 connected and WITHOUT any RS232 channel open, Basic area starts at 23755. (23813 if Interface1)

<--- RAMTOP.

This area reserved for you to define your own graphics.

<---Physical RAMTOP.

The above map applies when you switch-on Spectrum. The RAMTOP is set to 65368 (You can always check this by typing PRINT PEEK 23730+256*PEEK 23731). The last Byte of User Memory Area is just Below this RAMTOP...normally 65367. If you did a CLEAR 30000, then 30000 is the last address of the User memory area. (Obviously a CLEAR 65535 gives you an extra 168 Bytes of room). BEWARE. The top area of the User Memory Area is used to hold the Return

Addresses from GOSUBs and is also used by the ROM to hold the "stack" info. (This means the RETURN addresses from machine-code actions). This means you that you should never load Code programs which go within approx 100 Bytes of the RAMTOP. Hang on! What about JETSET WILLY? It uses a block of Code which starts at 32768 and is 32768 long. The last Byte is in 65535! This is done by the Basic doing a CLEAR 32767: LOAD "" CODE. This sets RAMTOP to be BELOW the Code block and therefore all the area used by the "stack", etc, is BELOW the Code.

Now lets consider the types of Machine-Code blocks we want to transfer. There are five categories:

1. A screen$. This has Start=16384. Length=6912
2. Code starting high in mem' & going over 65368.
3. Code starting low in memory....by low we mean starts below about 25000.
4. Code which Loads into System Variables area. This means between 23552 to 23755.
5. Excessively long blocks of code (I.E. Full 48K)

Obviously programs could be a combinations of the above. For example the game PINBALL by Sagittarian is 16384,16128. This covers the screen, the System Variable & continues into the Basic area. This is transferred and explained in later notes.

Codes which give no problems and which can be just Loaded then saved to Opus/Microdrive would be:

1. Screen$. (16384,6912)
2. Programs in Printer Buffer area. (23296,255)
3. Programs in UDG area. (65368 to 65535).
4. programs which have a Start address of over about 24500, even if they go upto very top of the memory.

JUDGE DREAD and FEUD, are over 40000 Bytes long but transfer easily. MERGEing in the Basic will reveal the RANDOMIZE USR value...must include a CLEAR which is 1 Byte below the start of the Code.

Transferring a screen$ is best done by one command LOAD "" SCREEN$: SAVE*"m";1;"name" SCREEN$ will load and transfer a screen picture without losing the bottom two lines of it.

Code which Load into the Sys Vars area cannot be simply Loaded and transferred as they will change. Easiest method is to simply load them higher in memory, Save them and ensure the Basic part quotes the correct location when re-loading. For example: If the Code was 23672,3 and called "x", we could Load it by LOAD "x" CODE 30000 then Transfer it by using; SAVE*"m";1;"x" CODE 30000,3. To make the Basic put it back into correct location, the Basic would be; 10 LOAD*"m";1;"x" CODE 23672 in this example. The "relocating" and "pulling" back to the correct address is a very powerful facility.

QUESTION? Why do you say Codes which start above 24500 are O.K? Why 24500?
ANSWER. When you have a Basic program in Computer, and then want to also Load in a Code, obviously the Code must not Load into the area being used by the Basic. In general, programs transferred to the Opus/Microdrive will be virtually all Code with a short Basic being used to Load in the Code part(s) from off the Disc/Cartridge. With the Microdrive the Basic area starts at 23813. You also have to allow approx 600 Bytes for the temporary Buffer. This means first address available for the Code, even if no Basic in machine at all, is a minimum of 23813+600=24413. Actually you have to allow an extra few hundred Bytes for the Spectrum workspace This means roughly that even if only using a short simple Basic Loader of one hundred Bytes at least, then the first "safe" address we could use for Code is 24713. (With the Opus this would be 24113, but for safety, use same figures as Microdrive). To fully ensure that the Code wont go on top of the Basic part, assume lowest address to Load in Code into the Basic area is about 25000. To help ensure we don't have trouble with the Basic, we always make the Basic as short as possible.

EXAMPLE GAME TRANSFER

If you have JETPAC, good. If not, do buy it or at least borrow it from a friend as it is the ideal program to demostrate the techniques. (Do NOT use Horace Goes Skiing as it is a trap for the unwary This will be dealt with later).

Load in 007SPY and press L. Play in JETPAC & write down the info it gives. YOU MUST DO THIS as there are two versions of this game, and figures differ slightly). Make a table as below;

NAME	TYPE	START	LENGTH	(BASIC AUTO)
JETPAC	BASIC	-	376	10
JPSP	CODE	16384	6912	
0	CODE	24576	8192	
1	CODE	23424	15	
2	CODE	23728	1	These last two are in
3	CODE	23672	2	Sys Vars area.

The method is basically to transfer all the Code Blocks to Opus/Microdrive, then make up a new Basic to Load each of the Code Blocks from off the 'Drive. only extra thing we need to know is how to make the program RUN. This is usually by a RAMDOMIZE, or PRINT USR value in the Basic. MERGE in the JETPAC Basic, LIST, and in last line is a PRINT USR 23424. Note this down for later use. Transferring to 'drive is very easy (making then RUN afterwards usually isn't). Each of the Code blocks is Loaded then saved to 'drive. BUT, if a block looks like it may give problems, load it HIGHER in memory. We do NOT want the Basic part as it is only several LOAD "" CODE to Load from off a tape. The method for Jetpac would be as follows:

a. LOAD "JPSP" SCREEN$: SAVE*"m";1;"JPSP" SCREEN$
   (Note that for a screen$ the keyword SCREEN$ can be used instead of putting CODE 16384,6912)
b. LOAD "0" CODE 30000:
   SAVE*"m";1;"0" CODE 30000,8192
c. LOAD "1" CODE 23424:
   SAVE*"m";1;"1" CODE 23424,15
d. LOAD "2" CODE 30000:
   SAVE*"m";1;"2" CODE 30000,1
e. LOAD "3" CODE 30000:
   SAVE*"m";1;"3" CODE 30000,2

Part "0" doesn't really need Loading higher if the Spectrum is empty, but its good practise to always temporarily move higher is a "low" address. Note that we Loaded part "1" into correct location. As this is the Printer Buffer it will be O.K.

Parts "2" and "3" are normally loaded into the System Variables area, so we have to move them higher to Load/Save them. The above method will do all of the older Ultimate games. You do NOT have to use 30000, but it is an easy value to remember. If the part to transfer is long, then could use a lower value. Remember that if the last Byte of the block would approach near to the RAMTOP, then you MUST do a CLEAR xxxx, with xxxx being at least 1 Byte below address you will Load the Block into. In above example we could use a CLEAR 29999, but as the parts are all very short it is not necessary. In theory, we only need now to make up a simple Loader for these Block, BUT, the part "0" is a bit low and likely to overwrite the Basic we have to use. The method is to Load this offending part Higher in memory, then when all Loaded in, and just before the RANDOMIZE USR to start the game, move it back to correct address. Doing this move by Basic would be slow, & may give problems. A super fast Machine-Code routine to do this "Block Move" is on the 007 MULTISPY near end of side 2 (called "007MOVER"). Make up the Basic by first Loading 007MOVER & reply to video questions:

MOVE FROM?	Type in 30000 as the part "0" we want to move will be in 30000 onwards.
MOVE TO?	Type in 24576 as this is the correct location for this part "0"
LENGHT?	Type in 8192 as part "0" is 8192 Bytes
USR?	Type in the RANDOMIZE or PRINT USR value we found in the Basic. (If a game does not have a USR value, then enter 0).

The 007MOVER will now display the values entered with the letters A to I with numbers against then, and with Line 9999 Listed. Edit down Line 9999 and change the letters A to I for the values given. Press Enter when all done, to put back the line. Delete line 10 as it is only there to calculate the values for the A to I and we have finished with it. Now we add in the rest of the Basic we require, and our Basic will appear thus;

```
10 LOAD*"m";1;"JPSP" SCREEN$:
   LOAD*"m";1;"0" CODE 30000:
   LOAD*"m";1;"1" CODE 23424:
   LOAD*"m";1;"2" CODE 23728:
   LOAD*"m";1;"3" CODE 23672
9999 DATA VAL "33",48,117,VAL "17",0,96,VAL "1",
   0,32,VAL "195",128,91: FOR X=VAL "65500" TO
   VAL "65511": READ Z: POKE X,Z: NEXT X: LET X
   =USR VAL "65500"
```

To ensure Basic is as short as possible, all the Loads are in one long line.

This Basic is now ready to be Saved to 'drive by;

```
SAVE*"m";1;"JETPAC" LINE 10
```

Note that to Save Bytes the 007MOVER uses VAL on the numbers. To help ensure Basic is as short as possible, all the LOADs are in one line.

When you enter LOAD*"m";1;"JETPAC" the Basic as above is Loaded from 'drive and line 10 Loads in the Code parts, and Loads part "0" higher in mem'. Line 9999 then Block Moves this part (instantly) down to correct location, then does a Jump to the original USR value. (The last three Bytes of the DATA statement are 195,128,91. The 195 is a Z80 machine-code command to Jump to the address formed by the next two Bytes).

The aforementioned method applies to virtually all the Ultimate games, plus very many others. SABRE-WULF is an oddment in that the block of Code which would need temporarily loading higher, is fairly long. This means that instead of moving it to 30000, you could use say, 28000. Note also that the DATA statement has to be put somewhere in the memory. The 007MOVER puts its DATA into 65500 onwards. It can be put ANYWHERE provided that the other Code doesn't wipe it out. With very long programs, the DATA could be put onto the screen area if necessary (by altering the FOR loop and the USR value at the end).

- - - - - - - - - -

SINGLE PART CODE

As mentioned previously, an example of a one part machine-code game (the game had to be Loaded by

LOAD "" CODE) is PINBALL by Sagittarian Software. This is an old game, but is STILL the best Pinball game available. This Code is 16384,16128. This is simply Loaded and transferred by entering;

```
LOAD "" CODE 30000:
SAVE*"m";1;"PINBALLc" CODE 30000,16128
```

Load in the 007MOVER program & reply to questions:

MOVE FROM?	Type in 30000
MOVE TO?	Type in 16384
LENGTH?	Type in 16128
USR?	Type in 0 as there isn't a RAND'.

Delete the Line 10 of the 007MOVER, then add in;

```
10 LOAD*"m";1;"PINBALLc" CODE 30000
```

Save this Basic to 'drive by entering;

```
SAVE*"m";1;"PINBALL" LINE 10
```

My personal method is always to put a small c at end of Code parts so that a CAT command would show which blocks are Code.

- - - - - - - - - -

HEADERLESS-FILES

These CANNOT be transferred! Dissapointed? What we can do is to convert the Headerless program into a normal Bytes type program, then we can transfer it. The main problem here is that we have to find out the address the program is supposed to Load into. As the program is Headerless, we cannot get this info from the Header! How does the Spectrum get this info normally? When original tape is Loaded, the part on the tape BEFORE the Headerless-File contains the info necessary. This means we need to check thro' this previous part. Whilst this really should be done with a Dissassembler, near the end of 007 MULTISPY is a program called SEARCH. This is a pseudo Dissassembler made exclusively to look for the type of "Loader" used by Headerless-Files. With most games which have a Headerless part, this Headerless part IS the game, and all the rest can be ignored. (Still worth checking the Basic at it could possibly hold the RANDOMIZE USR value for the game.....very unlikely).

Method used is as follows:

a. Use the 007FILE program to put a Header onto the File.
b. Use 007SEARCH program to get the Start Address, Length and "RANDOMIZE USR" value.

After the above is done, the Headed File can be simply transferred to 'drive.
Once again the example is done using an old (but was a high selling program). It really is worth while getting hold of some older programs as they are best to experiment with to familiarise yourself with transfer methods. For Headerless-Files types, we will use CYRUS IS CHESS.

a. Load in 007FILE then press P as the File is obviously not very long, then play in the Headerless File. When all in, press Enter to Save out to a tape (with a Header on it).
b. Reset spectrum then load in the Code part which is just before the file by LOAD "" CODE 30000 It does NOT matter what the correct address is as we are only going to look for the "Loader" for the Headerless part.
c. Load in the 007SEARCH. It will ask for address to start search from. As we Loaded the Code into 30000, we enter 30000. The SEARCH program looks thro' the Code for the grouping of coding which is a Loader for Headerless. When it finds this it will print it as:

```
30030  LD IX 24576  THIS IS START
30034  LD DE 16384  length
30037  LD A  255
30039  CALL  1366
30042  JUMP  24576  THIS IS "RUN"
```

The message shown WILL be clearly printed on the screen. Numbers on the left are the locations it found these, but these can be ignored. What the example above tells us is that the "Headerless-File" has a Start of 24576 a Length of 16384 Bytes and to Run it we have to do a RANDOMIZE USR 24576. It isn't necessary to Load the program Higher as the Basic used to Load it off the 'drive will be extremely small. (We've already put a Header onto the File). The new "Headed" program is now Loaded and put onto 'drive by;

```
LOAD "" CODE 24576:
SAVE*"m";1;"CHESSc" CODE 24576,16384
```

The Basic to Load it would be simply;

```
10 LOAD*"m";1;"CHESS" CODE 24576: RANDOMIZE USR
                                  24576
```

LONG PROGRAMS

Long programs, which means those aroung 40K, are only awkward if they start below about 24500. As stated previously, JUDGE DREAD is 40546 Bytes, yet transfers easily as it starts at location 24736. SCRABBLE or HOBBIT are awkward as they start too low in memory. SCRABBLE starts at 24400. Note that there are THREE versions of SCRABBLE, and FOUR versions of HOBBIT; each has different length. By Chopping into TWO blocks, using the MULTICHOP tape, the game can be transferred to Drive. SCRABBLE will be used as an example. If your version of SCRABBLE is the same as this one, (the long block was 24400,41135), then simply use the values given. If different, then adjust the values as necessary. With games having long blocks of 40K or there about, only the long block is the game, and the rest can be ignored. The method used is to split the long block into two parts. Part 1 being 4000 Bytes and Part 2 being the remainder. Part 2 is then loaded into correct area, and Part 1 is Loaded onto the screen. When all Loaded the part on the screen is Block Moved to correct address and program then RUNs. The screen will look a mess initially, but rapidly becomes normal. As usual with most games, only the last Code Block is really the game, so the screen$ and basic part(s) can be ignored after you've MERGEd them in to check for any RANDOMIZE USR values.

Method using SCRABBLE as an example;

1. Load "MULTI" and menu of options will appear.
2. Wind SCRABBLE to be just at beginning of the Header for the long Code block, last on the tape, Press **3** to select 4000+REM then play in the long block of SCRABBLE code. When loaded, put a blank tape in recorder, set to record, then press enter key. Two blocks will be Saved out with a gap between then. First block will be the first 4000 of Code fed in, and second will be the remainder, this is 37135 Bytes long. When the Save is complete, the Spectrum will Reset.

We now have the Code split into two parts & will use a Basic to Load the long part to correct loc' in memory, and put other temporarily onto the screen. Transferring the two parts to 'drive can be done several ways, but care is needed with part 1 as it is low in memory. best method is:

1. Enter LOAD "" CODE 30000 then play in all of part 1 (which is 4000 Bytes long).
2. Save to 'drive by
   SAVE*"m";1;"SCRABBLE1" CODE 30000,4000
3. Enter CLEAR 28399: LOAD "" CODE 28400
   (Note that we have to use a CLEAR as this block goes upto very top of memory).
4. Save to 'drive by
   SAVE*"m";1;"SCRABBLE2" CODE 28400,37135

Now ready to start making up the Basic and will have to use the 007MOVER program to move the 4000 Bytes we temporarily put onto the screen area.

a. Load in 007MOVER and answer the questions:
   MOVE FROM? Type in 16384
   MOVE TO? Type in 24400
   LENGTH? Type in 4000
   USR? Type in 24400

(You had remembered to MERGE in the Basic on its own first to get the RANDOMIZE USR value)?

b. Edit down line 9999 and substitute the A to I for the values it gives on the screen.
c. We CANNOT allow the FOR loop in line 9999 to put the DATA into 65500 as part of the program uses this area. A good place to put it would be on the screen at location 20480 (start of bottom third of the screen). This means change the FOR loop to be; FOR X=20480 TO 20493 and change the LET X=USR value to be 20480.
d. Delete line 10 as it has done its job.
e. Add in the extra lines thus:
   10 CLEAR 28399
   20 LOAD*"m";1;"SCRABBLE2" CODE 28400
   30 LOAD*"m";1;"SCRABBLE1" CODE 16384
   (next line would be line 9999 of 007MOVER)
f. Save to drive by SAVE*"m";1;"SCRABBLE" LINE 10

Note that we use a CLEAR which is 1 Byte below the start of the long Block as it goes up to the top of memory. By making the "screen" 4000 Bytes Load last, the "rubbish" on the screen is there for the minimum time. The game will now load in about 15 seconds instead of the usual 4 minutes. The only action you will NOT have if the ability to save a part played game as we CANNOT return back to Basic after the block move as the 4000 Bytes are sitting over part of the Microdrive memory map AND over our Basic. As we jump to the "RANDOMIZE USR" value whilst still in machine-code (by the DATA statement action) we can do this safely.

- - - - - - - - -

FULL 48K PROGRAMS

You CANNOT save a full 48K in one Block. The method used is to again use the MULTICHOP, but this time chop-off the screen$ part and split the rest into 4000 Bytes and the remaining 38240 Bytes Note that a genuine 48K game will start at 16384 which is the start of the screen area. The method for ALL full 48K games is identical and would be;

1. LOAD "MULTI" and select Option 4. This will chop-off the screen and split the rest into two blocks. Block 1 would be 4000 Bytes & Block 2 would be 38240 Bytes long.
2. Play in the very long Block of the game; MULTI doesn't care if it is Headerless type or not as it ALWAYS saves out blocks with Headers on them.
3. When all Loaded in, pressing Enter will start the save to a blank tape.
4. Load and transfer the two blocks to 'drive by;
   a. LOAD "" CODE 30000:
      SAVE*"m";1;"name1" CODE 30000,4000
   b. CLEAR 27295: LOAD "" CODE 27296
      SAVE*"m";1;"name2" CODE 27296,38240
5. Reset Spectrum. Load in 007MOVER and answer the questions:

MOVE FROM?	Type in 16384
MOVE TO?	Type in 23296
LENGTH?	Type in 4000
USR?	Type in 0

Edit down line 9999 and substitute the A to I with the values given on the screen. The FOR loop in line 9999 has to be changed as the Code block will be using the top of the memory. The FOR loop part is changed to be;
FOR X=20480 TO 20493: READ Z: POKE X,Z: NEXT X: LET X=USR 20480

6. Delete line 10 as it is no longer required. Add in the following Basic lines;

```
10 CLEAR 27295
20 LOAD*"m";1;"name2" CODE 27296
30 LOAD*"m";1;"name1" CODE 16384
```

(Next is the line 9999 of 007MOVER)

7. Save to Drive by; SAVE*"m";1;"name" LINE 10

AWKWARD PROGRAMS

HYPER LOADERS

HYPER load programs are very difficult to transfer as the method on the tape is to Load the program by Loading in a block of Code, which is a copy of the Sinclair ROM Loader, with a few modifications. The 007SPY can be used to reduce the Fast parts down to normal speed. (This is done by following the instructions for 007SPY to Load in Fast parts, but press N before saving out). The program can be easily transferred to 'drive, using previously mentioned techniques, but making it RUN is very difficult as it would be necessary to use a Dissassembler to hopefully find the RANDOMIZE USR value....what you have to find is the address it jumps to after it has loaded the Fast part. In majority of cases this will be impossible to find as there are so many way which could be used. At the end of these notes are several programs which have been "cracked" and the transfer method. We regret that these would only apply to the actually programs named. By using a Dissassembly & looking at the parts before the Fast parts, and comparing with the Sinclair ROM Loader routine (which is at location 1366 to 1542), you will eventually be able to recognise the Loader program.

MACHINE CODE LOADERS (and double MOVER)

A few games use a simple routine to load in parts which appear to be normal. One example is HORACE GOES SKIING. Obviously using a Dissassembler would be helpful. To load a Headerless File by machine code....explained earlier, requires that the IX, DE and A register in the Z80 chip are preset to the Start, Length & Code type. If loading a normal block, then also have to load in the Header which isn't really wanted. Usual method if to Load the Header 17 Bytes into location 0, which means it Loads into the ROM! This is impossible and really means "throw it away". Dissassembling the start of HORACE (the part called "ski" Code) gives:

```
LD IX,0          This is throw away the Header, which
LD DE,16         they have made 16 Bytes instead of
XOR A            17. The XOR A is the same as LD A,0
SCF              which is the "code" for a Header.
CALL 1366        CALL 1366 is a GOSUB to ROM Loader.
-----------
LD IX,23760      The last block is really loaded into
LD DE,694        location 23760 and is 694 Bytes long
LD A,255         The JP 24432 tells us that the game
SCF              actually starts at location 24432.
CALL 1366        Note also that 23760 is far too low
JP 24432         even without an Interface 1.
```

When the MOVER is added to the Basic, the Basic gets too long and it now means another block (which loads in 24576) is then too low and also require the MOVER. Ideally we require a double MOVER in machine code, but for non-machine code people, we will use the 007MOVER twice. (Bet you thought HORACE transfer would be easy)? Method used will be to Loads the two Code parts of HORACE high in memory, then save to 'drive as follows:

1. Load in the two parts required by entering;
   LOAD "ski" CODE 30000: LOAD "m" CODE 40000
2. Transfer to 'drive by;
   SAVE*"m";1;"HORACE1" CODE 30000,8050:
   SAVE*"m";1;"HORACE2" CODE 40000,694

Next we will use MOVER twice...explained later.

3. Load 007MOVER and STOP it. To save time later, Edit down line 9999 and make it Line 9990. This gives us two identical lines at 9990 and 9999. Press RUN then answer the question thus;

MOVE FROM?	Type in 30000	This is being
MOVE TO?	Type in 24576	used to Move the
LENGTH?	Type in 8050	first Block so we
USR?	Type in 0	tell it no USR.

4. Edit down line 9999 and substitute the A to I. BEFORE putting the line put back, change the line number to 9990.
5. Press RUN again and this time answer questions;

MOVE FROM?	Type in 40000	This is for the
MOVE TO?	Type in 23760	second block so
LENGTH?	Type in 694	put in the true
USR?	Type in 24432	USR value.

6. Edit down line 9999 and substitute the A to I, then press enter. Delete line 10 as not needed anymore. We now have two MOVERs set up.

IF AN OPUS, GO TO NOTE 8 AND IGNORE NOTE 7.

7. Enter MERGE "007RECLAIM" (don't use Load as it would wipe out our Basic), play in 007RECLAIM. RECLAIM is in line 9998. We have to shuffle the lines so that it is BEFORE the MOVERs. Do this by Editing down line 9998, and change it to be line 9000 (then Delete line 9998).
8. Add in the following few lines;

```
10 LOAD*"m";1;"HORACE1" CODE 30000
20 LOAD*"m";1;"HORACE2" CODE 40000
```

10. Save to 'drive by;

```
SAVE*"m";1;"HORACE" LINE 10
```

Complete action is lines 10 & 20 Load in the Codes higher in memory. (If a Microdrive, line 9000 will then RECLAIM the Map area or else the first move would overwrite part of Basic). First MOVER will move first Block, then second MOVER moves second block and starts the game. This is complex, but is unfortunately necessary for this game.

MICRODRIVE RULE: For safety, best to ALWAYS use the RECLAIM routine BEFORE the MOVER(s).

MICRODRIVE NOTE

Several of the previously mentioned routines pull programs over the Microdrive Map area. This does not matter if we stay in machine-code and immediately do a jump to start the game running. With games we usually jump into the game, which is machine-code, and never return back to Basic. With a game/program which is all in Basic, with a few DATA statements and/or a few machine-code parts in REM lines, these will give problems as all addresses are raised by 58 Bytes due to the extra Microdrive Variables which are switched in. Ideal method for such Basics is to Load program from off Microdrive, then "reclaim" these 58 Bytes as we no longer require then in the game/program. On the 007 MULTISPY, at end of side 1, is a short program called 007RECLAIM. This can be used, with care, to switch out the Microdrive properly by the action of reclaiming these 58 Bytes. For safety, ALWAYS use it BEFORE any MOVER routines. (An example is HORACE on page 24). Try this demo of RECLAIM.

1. RESET Spectrum by pressing Reset Button, or by powering OFF then ON...or by entering RUN USR 0
2. Enter this single line;
   10 PRINT PEEK 23635+256*PEEK 23636
3. RUN, and it'll print 23755 on the screen. The contents of the two Bytes 23635/6 ALWAYS hold the start address of the Basic User area.
4. Enter CLS ∉ This is the simplest Microdrive command and ensures in Microdrive mode and the extra 58 Bytes are switched in. RUN, and this time 23813 is printed on screen. This shows that whenever Spectrum knows Microdrive is present, the Basic is moved up by 58 Bytes.
5. Enter MERGE "007RECLAIM" then play in the last program on side 2 of MULTISPY. LIST and you'll see that my RECLAIM is in line 9999.
6. Your line 10 is still there. RUN, and again the screen shows 23813. Do NOT RUN again as you can not reclaim twice. If you enter as a direct command; PRINT PEEK 23635+256*PEEK 23636 then 23755 will be printed. This shows that the line 9999 action has reclaimed the 58 Bytes.

Care has to be taken when adding this line to your programs as you may have DATA statements in YOURS.

After obeying my DATA the next DATA it would try to read when it meets a READ command, would be AFTER my line. The best way to use 007RECLAIM is as follows:
Suppose your Basic normally starts from line 10. Load in your Basic, and Stop it. MERGE in the 007-RECLAIM. For safety, change the beginning of line 9999 so it is; 9999 RESTORE 9999: DATA etc, etc Also put at the very end of the line 9999 :RESTORE: GOTO 10
Save complete program to Microdrive by;

SAVE*"m";1;"name" LINE 9999

Action now is that when you reload this program off Microdrive, line 9999 reclaims the Microdrive Map area, then jumps to start of your program. Note that the FOR loop puts my DATA into top part of memory, but this could be moved elsewhere if it clashes with DATA in your program.

007VAL

This program will save Bytes in Basic programs by changing all numbers to be VALed. Changing GOTO 10 to GOTO VAL "10" would save three Bytes. In very long programs, the saving can be upto about 7000! 007VAL can be loaded anywhere in memory, then you simply RANDOMIZE the address you loaded it into. Usually safest to put it on the screen. EXAMPLE:

1. Load your Basic program and Stop it.
2. Enter LOAD "007VAL" CODE 16384 then play in the 007VAL.
3. Enter RANDOMIZE USR 16384 and after a short pause, the O.K. message appears. LIST and all your program has been VALed.

007RECLAIM

Use to "reclaim" the Microdrive Map area. See the top part of this page for example of use.

007FILE

This program will put a Header onto ANY Headerless program.

1. LOAD "007FILE" when loaded TWO option appear.
   a. P=PROGRAM FILE. Use this Option P if your Headerless program is known to be short OR if when you Load normally this part does not include the screen draw-up.
   b. S=SCREEN+PROGRAM FILE. Use this Option S if your Headerless program includes the screen$ when Loading normally.
2. Press P or S to select Option required, then play in the Headerless part. Note that it may cause "rubbish" to appear on the screen as this program uses the screen as work space.
3. When all Loaded, place a blank tape in recorder set to record, then press Enter key and your program will be saved out with a Header on it and will now have the title NEW HEADER.

Note that the Header didn't really know the correct location for your File so it says it has a Start address of 0, but will give the correct length. This means that if your File was the type which started on the screen, you could Load the saved out program by; LOAD "" CODE 16384. Note also that when using option S, the first few hundred Bytes will be corrupted. This will appear as just a few extra dot and lines on the screen. Doesn't matter as we would chop-off the screen before transferring to drive.

==========

TRANFER PROGRAM 1,2,3

See notes on page 9.

007SEARCH

Used to find the info from Headerless File Loaders. The Loaders MUST be on your tape BEFORE the File itself.

1. Load in the Code part before your File by;
   LOAD "" CODE 30000
2. Enter LOAD "007SEARCH" then play in 007SEARCH.
3. Enter the address to start Searching from. This would be 30000 if you loaded your Code as per note 1 above.

When found the info will be printed in plain English. Example print could be;

```
30100   LD IX 34567  THIS IS START
30104   LD DE  5555  length
30107   LD A   255
30109   SCF
30110   CALL   1366
30113   JUMP   35000 THIS IS "RUN"
```

This tells you that your Headerless File is intended to load into address 34567 and is 5555 Bytes long. The RANDOMIZE USR value to start this program running, in this example, is 35000.
Note the order may differ slightly from the above.

4. If it prints NOTHING, then it didn't find the Loader. This can happen if the Loader is in a Basic program (in a REM or in a DATA statement)
5. IF IN A REM
   The simplest solution is to Load in the Basic with the REM in it, then MERGE in 007SEARCH and enter GOTO 9999 and tell it to start Search from address 23755. If this is not possible, then Load in the Basic and Stop it (use the 007SPY if necessary to remove any auto-start). To avoid complications of Basic moving up when doing a SAVE if a Microdrive, save to a TAPE as follows: SAVE "x" CODE 23755, L with L being the length of the Basic. Reset Spectrum, then reload this Tape by LOAD "" CODE 30000 then Load in 007SEARCH and tell it to start Search from 30000.

b. IF IN A DATA STATEMENT
   List the Basic and find the FOR loop which

moves the DATA into memory, and change it to move the DATA into 30000 onwards. Note that if there are 20 values in the DATA statement, then you'd change the FOR loop to be;
FOR x=30000 TO 30019: etc,etc.
Find the line which contains the RANDOMIZE USR which normally "RANDOMIZEs" this DATA after it had been moved, and change the RANDOMIZE USR xxxxx to be STOP. RUN the FOR loop line so that the DATA is moved into 30000 onwards, then Save this Data to tape (or Drive) by;
SAVE "x" CODE 30000,L with L being number of Bytes to be saved.
Reset Spectrum. Load in the tape just Saved by LOAD "" CODE 30000. Enter LOAD "007SEARCH" and load in the SEARCH program. Tell it to start Searching from 30000.

==========

007MOVER

This is a machine-code Block Mover routine which could "move" 64K in less than 1 second. The MOVER has to be told addresses Moving FROM, TO and the LENGTH. In order to enable us to Block Move programs and RUN them without exiting from Machine Code, the "RANDOMIZE USR" value for a game can also be entered. If no RANDOMIZE USR value, then enter 0. Try this Demo Example:

1. Load in 007MOVER When Loaded, it asks you the FROM, TO, LENGTH & USR values. Answer these as follows (for this example):

MOVE FROM?	Type in 30000
MOVE TO?	Type in 16384
LENGTH?	Type in 6912
USR?	Type in 0

   The USR value would normally be the RANDOMIZE USR value for the game which is usually found in the Basic part. If none, them enter 0.
2. Program will now display the values you entered split into A to I and Lists line 9999.

3. Edit down line 9999, by pressing EDIT key, or by pressing CapShift & 1 Key. Substitute the letters A to I in line 9999 for the values displayed, then press Enter to put the line back. Delete line 10 as it is only there to calculate the values for us, and we no longer need it.
4. For this Demo example, add in a line 10, thus;
   10 LOAD "" CODE 30000: PAUSE 0
5. For safety save this Basic to drive by;
   SAVE*"m";1;"demo" LINE 10
6. RUN program, then play in screen$ from a game tape. When Loaded, screen will be blank as the picture has been Loaded into 30000 onwards.
7. Press Enter key, and picture will instantly be Block Moved form the 30000 area to the screen.

MULTICHOP TAPE

This tape contains FOUR programs to chop your long games down to size and/or remove screen$ part of a full 48K game. All programs work with Headed or Headerless-Files, and ALWAYS Saves out blocks with Headers on them.

Load "MULTI" and menu of options will appear.

1...Chop off 1st 6912
2...Save the 1st 6912
3...Split 4000 + REM
4...Chop-Split (1 & 3)

OPTION 1. Will remove the screen part from programs which load onto screen and continue Loading more all in one block. Useful for shortening games to save Bytes on Disc/Cartridge.

OPTION 2. Saves just the first 6912 Bytes. Could be used to save out just the screen$ part of a block.

OPTION 3. Splits whatever is fed in, into two parts. Part 1 would be 4000 Bytes and part 2 would be the remainder.

OPTION 4. Combines the actions of 1 & 3 above. Very useful on full 48K programs as it Chops-off the screen and splits the rest into two managable blocks.

NOTE 1: Option 2 is the only one to always restart Multichop after use. Due to the others using the the screen as workspace, and overwriting the Vars area, after the Saves completed, Spectrum Resets.
NOTE 2: MULTICHOP will NOT work on programs which are less than 6912 Bytes long. If less than 6912, MULTICHOP will reject it and restarts.

We regret that it just isn't possible to get a tape capable of making Back-Up copies of All games and it never will be possible to transfer everything, altho' the ZX-GUARANTEED products are far more sucessful than any other. (We also supply our products at lower prices as our products sell World wide on recommendation alone).

The Transfer routines which follow are for some of the more awkward programs which require special techniques.

```
**************************************************
*  Note that as the Transfer programs/techniques *
*  are semi-automatic methods, it means that the *
*  the 007 MULTISPY and MULTICHOP will also be    *
*  able to Transfer programs to the SPECTRUM+3    *
*  Disc system when it becomes available. Since   *
*  this will be based on the Amstrad Disc unit,   *
*  and ZX-GUARANTEED also produces items for an   *
*  Amstrad 6128 Computer, (Disc system), we are   *
*  fully aware of the Disc requirements.          *
**************************************************
```

AWKWARD TRANSFERS

PROBLEM: Games which require POKEs to be added to enable the transfer, will have different addresses if a Microdrive. If these addresses differ, these will be stated in the routine. The use of CLS # before Loading is not necessary for OPUS, but has been included to ensure that if an Interface 1 is present, the extra variables are switched in.

Game transfers which use the Back-Up copy made by the DE-PULSER program will have the main blocks as Headerless-Files. These will require Headers to be put onto then by using the 007FILE or could make "False Headers" for them as follows: To make a False Header for a CODE program which is 6912 Bytes long;
Enter SAVE "x" CODE 0,6912 and Save out to a blank tape, but STOP tape IMMEDIATELY the computer pauses. This means IMMEDIATELY after the first lot of coloured bars on the screen clears. This really means we have saved out a HEADER for 0,6912, and can be used to load in any Headerless File made by DE-PULSER if it is 6912 Bytes long. This False Header can Load in ANY Headerless screen$ as a screen$ is 6912 Bytes long, by entering;
LOAD "" CODE 16384 then play in this "x", then playing in the Headerless screen$

RAMBO

Need to make a Back-Up copy tape using DE-PULSER.

1. Make two False Headers. First for 31337 Bytes, and second for 11628 Bytes.
2. Load 31337 False Header by; LOAD "" CODE 26384
3. Play in the first long Headerless File of your RAMBO Back-Up tape made by DE-PULSER.
4. Save to Drive by;
   SAVE*"m";1;"RAMBO1" CODE 25500,22211
5. Load 11628 False Header by; LOAD "" CODE 30000
6. Play in the second long Headerless File of your RAMBO Back-Up tape.
7. Save to Drive by;
   SAVE*"m";1;"RAMBO2" CODE 30000,11628

8. Type in this Basic Loader:
```
10 CLEAR 25499: LOAD*"m";1;"RAMBO1" CODE 25500
20 LOAD*"m";1;"RAMBO2" CODE 53396
30 RANDOMIZE USR 26368
```
9. Save to Drive by; SAVE*"m";1;"RAMBO" LINE 10

HIGHWAY ENCOUNTER

Need to make a Back-Up copy tape using DE-PULSER. This Back-Up made by DE-PULSER is different in that it actually DOES copy the short "clicking" part after the Basic. Make a False Header for Code of 39168 Bytes.

1. Set the Back-Up to be just at the start of the last long Headerless-File.
2. Enter CLEAR 26383: LOAD "" CODE 26384 then play in the False Header you made, then play in the Back-Up tape.
3. Save the two blocks to Drive by;
```
SAVE*"m";1;"HE1" CODE 33296,3000
SAVE*"m";1;"HE2" CODE 36296,29256
```
4. Type in this Basic Loader, which includes its own MOVER type routine;
```
10 CLEAR 65535: LOAD*"m";1;"HE1" CODE 26296
20 LOAD*"m";1;"HE2" CODE 55808
30 DATA 33,0,128,17,0,91,1,184,11,237,176,6,
   128,237,176,95,128,237,79,195,225,176
40 FOR X=55555 TO 55575:READ Z:POKE X,Z:NEXT X
50 RANDOMIZE USR 55555
```
5. Save to Drive by; SAVE*"m";1;"HE" LINE 10

UNDERWURLDE

This routines works by altering the way original Loads in. This means you do NOT have to make a Back-Up copy first.

1. Use the 007SPY (option A) to get a STOPPED copy of the Basic, or use the method explained on page 11.
2. Enter CLS ≠ : LOAD "" then play in the Stopped Basic part of UNDERWURLDE.
3. If a MICRODRIVE, enter these POKEs, etc
```
Enter  POKE 24849,251: POKE 24850,207
Enter  RANDOMIZE USR 24798
```

4. If an OPUS, enter these POKEs, etc
   Enter POKE 24791,251: POKE 24792,207
   Enter RANDOMIZE USR 24740
5. Enter POKE 62410,251: POKE 62411,201
   Enter RANDOMIZE 62374
6. Play in all the rest of UNDERWURLDE after its first Basic.
7. Save to Drive by;
   SAVE*"m";1;"UNDERWc" CODE 26610,34820
8. Type in this Basic Loader;
   10 LOAD*"m";1;"UNDERWc" CODE 26110
   20 RANDOMIZE USR 26610
9. Save to Drive by; SAVE*"m";1;"UNDERW" LINE 10

KNIGHTLORE

Does not use a Back-Up copy as this routines works by altering the way original Loads in.

1. Use the 007SPY (option A) to get a STOPPED copy of the Basic, or use the method explained on page 11.
2. Enter CLS # : CLEAR 60000: LOAD "" then play in the Stopped Basic part of KNIGHTLORE.
3. If a MICRODRIVE, enter POKE 24849,195
4. If an OPUS, enter 24791,195
5. Enter POKE 63872,251: POKE 63873,51:
   POKE 63874,51: POKE 63875,201:
6. If a MICRODRIVE, enter RANDOMIZE USR 24802
7. If an OPUS, enter RANDOMIZE USR 24744
8. Press NEW then Enter.
9. Enter POKE 62410,251: POKE 62411,201:
   RANDOMIZE USR 62374
10. Play in all the rest of KNIGHTLORE tape.
11. Save to drive by;
    SAVE*"m";1;"KNIGHTc" CODE 24832,30720
12. Type in this Basic Loader;
    10 LOAD*"m";1;"KNIGHTc" CODE 24832
    20 RANDOMIZE USR 24832
13. Save to Drive by; SAVE*"m";1;"KNIGHT" LINE 10

TORNADO LOW LEVEL

The older version can be transferred by using the 007FILE to put a Header on the Headerless-File,and transfer similar to SCRABBLE method. This routine is for the later version (the Basic is "invisible"

on this version) which loads Headerless part into 16352 onwards. Altho 49052 Bytes long, only 39104 Bytes are required AND a POKE must be added or else it will crash when reloaded.

1. Type in then RUN the following, then play in the long Headerless part only; (ignore error message after Loading)

```
 10 CLEAR 65535
 20 DATA 221,33,224,63,17,156,191,62,255,55,205,
    86,5,207
 30 FOR X=65430 TO 65443: READ Z: POKE X,Z: NEXT X
 40 RANDOMIZE USR 65430
```

2. Save to drive by;

```
SAVE*"m";1;"TLLc" CODE 26300, 39104
```

3. Type in this Basic Loader;

```
 10 CLEAR 65535
 20 LOAD*"m";1;"TLLc" CODE 26300
 30 POKE 65435,153: RANDOMIZE USR 36258
```

4. Save to Drive by; SAVE*"m";1;"TLL" LINE 10

CHUKIE-EGG (original version)

Not really difficult, except the Headerless part is really TWO Headerless parts with no gap between them. First File is actually two Bytes long.

1. Type in this program;

```
 10 RESTORE 10: DATA 221,33,0,0,17,2,0,55,62,255,
    205,86,5,221,33,20,130,17,180,70,55,62,255,
    205,86,5,201
 20 FOR X=30000 TO 30026: READ Z: POKE X,Z: NEXT X
 30 RANDOMIZE USR 30000
```

2. Set CHUKI tape to start of the first Headerless part, then RUN above program and play in tape.
3. Save to Drive by;

```
SAVE*"m";1;"CHUKIEc" CODE 33300,18100
```

4. Type in this Basic Loader;

```
 10 PAPER 0: INK 0: BORDER 0: CLS:
    LOAD*"m";1;"CHUKIEc" CODE 33300
 20 POKE 23613,1: POKE 23614,0:
    RANDOMIZE USR 42000
```

The Loader discarded the 2 Byte File but this is required in the program....hence the two Pokes.

VALHALLA

The awkward part of this game is knowing which part we need to transfer.

1. Type in and RUN the following;
   ```
   LOAD "vmc" CODE 30000:
   ```
3. Play in the part called "vmc"
4. Save to Drive by;
   ```
   SAVE*"m";1;"VHALL1" CODE 30000,12185
   ```
5. RESET Spectrum. Enter CLS ≠ : MERGE "vdrive"
6. Play in the "vdrive" part. After Loading, it will take approx 2 minutes before O.K. appears.
7. If a MICRODRIVE, ALL the RANDOMIZE values need changing by making the Randomizes in lines as follows, to be;
   ```
   5010 RANDOMIZE USR VAL "23818"
   5016 RANDOMIZE USR VAL "23818"
   6050 RANDOMIZE USR VAL "23850"
   6255 RANDOMIZE USR VAL "23877"
   ```
8. Save to drive by; SAVE*"m";1;"VHALL2" LINE 9800
9. type in this Basic Loader;
   ```
   10 CLEAR 53349: LOAD*"m";1;"VHALL1" CODE 53350
   20 LOAD*"m";1;"VHALL2"
   ```
10. Save to Drive by; SAVE*"m";1;"VHALL" LINE 10

SHERLOCK

This again is mainly awkward in knowing which part we need for the game. Only the part called "p" is required. This has to be split into three parts & we'll do this by a different method to demonstrate an alternative way of splitting programs.

1. Enter LOAD "p" CODE 16384 then play in the part "p". This will start loading Bytes onto the screen area. STOP the tape IMMEDIATELY the colours start to appear (this means when colour block start on top part of the screen...don't worry as long as you stop tape before these reach bottom of screen).
2. Save to Drive by;
   ```
   SAVE*"m";1;"SHERLOCK1" CODE 16384,4000
   ```
3. Rewind tape to start of "p" again.
4. Enter CLEAR 27935: LOAD "p" CODE 27936 then play in all of the part "p".
5. Save to Drive by;
   ```
   SAVE*"m";1;"SHERLOCK2" CODE 31936,33600
   ```
6. Rewind tape to start of "p" again.

7. Enter CLEAR 44319: LOAD "p" CODE 44320 & play in all of "p" once again. Because this is being loaded so high in memory, the top part of the program will go over the top of memory and the last 4000 Bytes will go onto the screen. These are the Bytes we require.
8. Save to Drive by;
   SAVE*"m";1;"SHERLOCK3" CODE 16384,40000
9. This Basic Loader includes a MOVER routine:

```
10 CLEAR 27935
20 LOAD*"m";1;"SHERLOCK3" CODE 61536
30 LOAD*"m";1;"SHERLOCK2" CODE 27936
40 LOAD*"m";1;"SHERLOCK1" CODE 16384
50 DATA 49,127,93,33,0,64,17,128,93,1,160,15,
   237,176,195,64,160
60 FOR J=23296 TO 23312: READ A: POKE J,A:NEXT J
70 RANDOMIZE USR 23296
```

10. Save to Drive by;
    SAVE*"m";1;"SHERLOCK" LINE 10

Do please study this transfer method and try to calculate why "p" was loaded into the various addresses each time. Part "p" is 23936,41600

VU-FILE (With main Code 25088,5640)

This can be transferred to Drive basically by making it load the Code part BEFORE the Basic.

1. Enter LOAD "C" CODE 25088 then can play in VU-FILE from beginning as all parts except the one required will be ignored.
2. Save to Drive by;
   SAVE*"m";1;"VUC" CODE 25088,5640
3. RESET Spectrum and rewing the tape.
4. Enter MERGE "" and play in the tape. The Basic will Load but not run.
5. Delete Line 50 and change line 100 to be;
   100 CLS ## : DIM f$(32): LET A=29785: BORDER 1: GOTO USR 29721
6. Change lines 1005,1100 & 2000 by adding in the *"m";1; as necessary.
7. Save to Drive by; SAVE*"m";1;"VUB" LINE 100
8. Type in this Basic Loader;

```
10 LOAD*"m";1;"VUC" CODE 25088: LOAD*"m";1;"VUB"
```

9. Save to Drive by; SAVE*"m";1;"VUFILE" LINE 10

VU-FILE (With main Code 25088,5888)

This version requires the Basic reducing in length slightly.

1. Enter this Basic Loader;

```
 10 LOAD*"m";1;"VUC" CODE 25088: LOAD*"m";1;"VUB"
```

2. Save to Drive by;

```
SAVE*"m";1;"VUFILE" LINE 10
```

3. RESET Spectrum, then enter MERGE "" and play in the tape till O.K. message appears.
4. Delete line 50. Delete the INK 7 from Line 100.
5. Add the *"m";1; to the commands in lines 1005, 1100 & 2000.
6. Change line 7000 to be; (note the ' used here)

```
7000 PRINT "If gives ERROR CODE restart by";'"GO
     TO USR a": RETURN
```

7. Save to Drive by;

```
SAVE*"m";1;"VUB" LINE 100
```

8. RESET Spectrum. Enter LOAD "c" CODE 25088 and play in rest of the tape till O.K. appears.
9. Save to Drive by;

```
SAVE*"m";1;"VUC" CODE 25088,5888
```

Whilst every attempt has been made to include as many awkward oddment programs as possible, we have only printed those which have received the most requests for routines. we do have more, and if you require the routine for any not in this booklet, please forward an SAE and if we have the routine, or can get hold of it, we will gladly forward it.

STOP-REVEAL

A Hackers delight. Enables you to STOP & REVEAL most normal speed programs. (Fast or Pulsing programs can be reduced first to normal speed by using 007SPY or the DEPULSER programs on 007SPY or 007 MULTISPY tapes). Includes program which has successfully STOPPED full 48K games (tested on SPLAT, ISLAND and ANT ATTACK). Also includes a Header Reader and a Dissassembler so can even display Machine-Code on your screen. Includes how to use Dissassembler to find Start, Length & RANDOMIZE value for Headerless programs.

£5.95 Post free to U.K./BFPO/Ireland

---

007 SUPERFILE 2

and

LISTFILE 3

SUPERFILE can hold upto 500 Records (names and addresses). Each Record can be upto 200 Char's. Features on screen Editing as the info in being typed in. Superfast Machine-Code used for SEARCH (can find any one out of 500 in under 1 second), ALPHA-SORTs, etc. Prints out to ANY printer as Records or Labels. Print out can be by "sort", one character or string of characters.

LISTFILE is a single Line record system, ideal for holding record of games tapes, etc.

BOTH programs supplied on one cassette. Note that these can both print out to ZX, Alphcom, RS232 or Centronic printers.

BOTH CAN AUTO-TRANFER THEMSELVES ONTO M/D or OPUS

£6.95 Post free to U.K./BFPO/Ireland

G.A.BORKER
29 Chadderton Drive, Unsworth, Bury, Lancs.
Tel: 061-766 5712

MICRODRIVE-EXCHANGE (Inc OPUS)

This was a Microdrive (and OPUS) Newsletter club, which was ran for 2 years. A total of 24 issues had been printed. These are available at £5 for 12 issues, or £9 for all 24....these form a useful set of references notes for M/drive or Opus

Post free in U.K. Europe +£1. Elsewhere +£2

==========

LEARN MACHINE CODE

Trying to learn from a book is possible, but books don't answer any questions and usually are boring. The solution is to join the new Z80 Newsletter Machine Code Club which will be starting in January 1987. Method will be to teach machine code in a more interesting and practical format. The complete course will consist of regular Newsletters, plus initially a Dissembler and Loader tape. Course will be Based on the Spectrum & Amstrad computers. ALL addresses will be stated in Decimal as it is absolutely vital to know precisely location the code is going in to. To avoid wasting time explaining why we should have eight fingers on each hand, a seperate set of notes on Binary, Decimal and Hex will be sent with issue one. Final note: Of course you can write and ask questions on each issue if you are having problems. Membership for complete course is as follows:

£15 U.K. £19 Europe. £22 Elsewhere

Please state if a Spectrum or Amstrad 464/664/6128

All cheques/P.O./Bank Drafts payable to: G.BOBKER

==========

ZX-GUARANTEED

29 Chadderton Drive, Unsworth, Bury, Lancs.
Tel: 061-766 5712