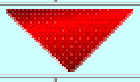
Format name: TR-DOS image format, Format creator: RamSoft

A TR-DOS disk contains 1 or 2 sides with 40 or 80 tracks per side and with 16 sectors per track. Each sector is 256 bytes long.

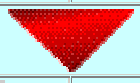
There are 4 different disk formats. The format is described in the disk specification (see below):

disk type:	sides:	total tracks:	total sectors:	total bytes:	reserved sectors:	reserved bytes:
double-sided, 80 tracks	2	160	2560 (0..2559)	655360	16 (0..15)	4096
double-sided, 40 tracks	2	80	1280 (0..1279)	327680	16 (0..15)	4096
single-sided, 80 tracks	1	80	1280 (0..1279)	327680	16 (0..15)	4096
single-sided, 40 tracks	1	40	640 (0..639)	163840	16 (0..15)	4096

The counting of the sectors includes the reserved sectors, that means that the first writeable sector has the index number 16.

TRD file format (all 4 disk types):				
Offset:	**Field type:**	**Length:**	**Description:**	**Additional information:**
0	directory	2048	file allocation table	the first track (16 sectors) is reserved and contains the FAT and the disk specification
2048	specification	256	disk specification	
2304	0-Byte -ARRAY	1792	filled with zero (filler)	
list of data sectors:				
4096	BYTE -ARRAY	256	16th data sector	The index of data sectors begins at sector #16. The data sectors contain the data of file bodies. The maximum count of sectors depends on the disk type (see table above).
4352	BYTE -ARRAY	256	17th data sector	
4608	BYTE -ARRAY	256	18th data sector	
				
???	BYTE -ARRAY	256	last data sector	

The TRD format uses the following sub-formats:

directory (2048 byes):				
Offset:	**Field type:**	**Length:**	**Description:**	**Additional information:**
+0	directory entry #0	16	header of 1st file	the directory always contains 128 entries, but the list ends when the first byte of the directory entry (= of the file name) is #0. Following directory entries are invalid and may contain random data.
+16	directory entry #1	16	header of 2nd file	
+32	directory entry #2	16	header of 3rd file	
				
+2032	directory entry #127	16	header of 128th file	

specification (256 bytes):				
Offset:	**Field type:**	**Length:**	**Description:**	**Additional information:**
+0	0-Byte	1	end of directory	must be 0 to indicate the end of the directory
+1	0-Byte -ARRAY	224	unused	filled with zeroes
+225	BYTE	1	first free sector (0..15)	representing the logical sector number of the next free sector on the disk. If the disk is full, the sector number is = count of sectors. logical sector=first free track*16+first free sector
+226	BYTE	1	first free track	
+227	BYTE	1	disk type	22: double-sided, 80 tracks 23: double-sided, 40 tracks 24: single-sided, 80 tracks 25: single-sided, 40 tracks
+228	BYTE	1	file count	0..128; the count of non-deleted files
+229	←WORD→ LSB MSB	2	free sectors	number of free sectors on the disk
+231	BYTE	1	TR-DOS ID	always 16
+232	0-Byte -ARRAY	2	unused	filled with 0
+234	CHAR -ARRAY	9	unused	filled with spaces (#32)
+243	0-Byte	1	unused	filled with 0
+244	BYTE	1	deleted files	number of deleted files on the disk
+245	CHAR -ARRAY	8	disk label	label name of the disk
+253	0-Byte -ARRAY	3	unused	filled with 0

The disk specification uses the following sub-blocks:

directory entry (16 bytes):				
Offset:	**Field type:**	**Length:**	**Description:**	**Additional information:**
+0	CHAR -ARRAY	8	file name	case-sensitive; if the first character is #0, then it's the end of the directory. #1 indicates a deleted file, which is still present on the disk.
+8	CHAR	1	file extension	character that decribes the file type: "B" = Basic program, "D" = DATA array (numeric or alphanumeric) "C" = CODE "#" = Print file (may be split into several sub-files with max. 4096 bytes each)
CASE	case 1: parameters of Basic program			
+9	WORD LSB MSB	2	progs+vars	length of program + variables area
+11	WORD LSB MSB	2	progs	length of program only
CASE	case 2: parameters of data arrays			
+9	WORD LSB MSB	2	param 1	unused
+11	WORD LSB MSB	2	data length	length of data array
CASE	case 3: parameters of code (byte array)			
+9	WORD LSB MSB	2	start address	
+11	WORD LSB MSB	2	code length	
CASE	case 4: parameters of print file			
+9	BYTE	1	extent no.	number of part of the print file, beginning with 0
+10	BYTE	1	unused	always #32
+11	WORD LSB MSB	2	print length	0..4096
				continuing:
+13	BYTE	1	file length	length of file in sectors
+14	BYTE	1	start sector (0..15)	represent the start sector, calculated as = start track*16+start sector
+15	BYTE	1	start track	

The file bodies are stored in the data sectors, beginning at sector #16.

Basic program and data array files have an addition at the end of the file.

The structure of the different file types is as follows:

file body of Basic program files:				
Offset:	**Field type:**	**Length:**	**Description:**	**Additional information:**
+0	BYTE -ARRAY	**[file length]**	data of the Basic program	the pure bytes that represent the basic (and if present, variables) data
+**[file length]**	CHAR CHAR	2	parameter 2 indicator	always #128 #170
+**[file length]**+2	WORD LSB MSB	2	autostart line number	0..9999

file body of data array files:					
Offset:	**Field type:**	**Length:**	**Description:**	**Additional information:**	
+0	BYTE -ARRAY	**[file length]**	data of the data array variable	the pure bytes that represent the content of the data variable	
+**[file length]**	CHAR CHAR	2	parameter 2 indicator	always #128 #170	
+**[file length]**+2	BYTE	1	unused		
+**[file length]**+3	BYTE	1	name of variable	bits 0..5:	1..26 meaning "a".."z"
				bits 6..7:	10b = numeric array 11b = alphanumeric array

file body of all other files:				
Offset:	**Field type:**	**Length:**	**Description:**	**Additional information:**
+0	BYTE -ARRAY	**[file length]**	data of the file	the pure bytes that represent the content of the file