

## SuperCharge & TurboCharge Decompiler Technical Notes

Technical notes.

### Checksums in Library files

The checksum is comprised of 40 bits. Split into two parts.

A 16 bit, byte count of the routine. And a 24 bit checksum of the bytes of the routine. And is shown as a 10 character hexadecimal number

example -

00100033C7

This routine contains 16 bytes (\$0010), and the 16 bytes make a checksum of (\$00033C7)

### Procedures and Functions

PROCedure SaveMe

Saves a copy of the program

FuNction GetString\$(ptr)

Returns a string of a standard QDOS string (word length then bytes of string) stored in memory at ptr

FuNction GetInt(ptr)

Returns a signed integer stored in memory at ptr

FuNction GetWord(address)

Returns an unsigned integer stored in memory at address

FuNction FLT(string\$)

Converts a 12 character (six byte) hex string into a floating point number

FuNction FLT\$(n)

Convert a Floating point number into a 12 character (six byte) hex string  
by PJW & Steven Pool V0.02 03/12/17

FuNction NameListLen(start)

Returns the number of the highest name list entry in the keywords section at the end of the compiled program

start is the address in memory of the keyword table

PROCedure EmptyPipe (channel)

Makes sure the pipe is empty. If it is not empty, it is reported to channel, the pipe is emptied displaying the contents of the pipe.

FuNction GetTOS\$

Remove and return the item at the rear end of the pipe.

PROCedure FlagLine (num\$)

If the line number, num\$ does not already exist in the Proc/Fun list, it is added.

FuNction CheckProcFun(num\$)

Checks to see if the line number, num\$ is in the Proc/Fun list. If it is, then create a DEFINE PROCedure/FuNction line.

Also checks for unresolved SElect's and IF's.

Returns

0 If the line number is not in the Proc/Fun list.

1 If its a multiline DEFINE PROCedure/FuNction.

2 If its a single line DEFINE PROCedure/FuNction.

PROCedure AssignArray(varNum,noElements)

Adds an array entry into the arrayElements array

FuNction GetElements(varNum)

Returns the number of elements for an array. varNum is the SuperCharge reference for the array.

PROCedure ListProcs

List the Procedure/Function line numbers to the log file

PROCedure ReadCodeArray (file\$)

Read in the code array from file\$. The Format is index number,hex word index value

E.g. 23,90FA - codeArray(23)=\$90FA

PROCedure DoDataLines

Create DATA statements. DATAarray(0) holds the number of RESTORE commands.

Other indexes are addresses of data starts.

PROCedure CheckELSE (ptr)

Check to see if an ELSE is needed. ptr is the current program position.

PROCedure StripIFStack

Strips the top item off the IFStack and ELSEStack string

FuNction FindLine(ptr)

Scan from the current program point (ptr) looking for the start of the next program line.

PROCedure LoadFile (file\$)

Load the file (file\$) and set base addresses initialBase, base, and filelength.

PROCedure CheckValues

Check values in file, and set various pointers.

#### PROCedure GetVersion

Get program version information and set marker variables.

#### PROCedure FindArrayElements

Scan variable initialization area looking for arrays, and filling in arrayElements.

#### PROCedure StartNewLine (linenumber) (tc)

Start a new listing line starting with the supplied line number.

#### PROCedure InitNewLine (tc)

Initialize line decompose variables.

#### PROCedure InitMain (tc)

Set up variables for decompiling.

#### PROCedure StatementSeparator (tc)

If line numbers are suppressed, start a new line. Otherwise output a colon (:) separator.

#### PROCedure CheckEndSelects (tc)

Checks for any outstanding END SElects to deal with

### Global variables

(sc) SuperDisCharge

(tc) TurboDisCharge

filelength	Length of executable file.
initalBase	Loaded base address, may be different to 'base' if Qemulator header found.
base	Start of the executable.
finish	End of the executable.
progOffset	Offset from start of executable to start of BASIC program area.
prog	Running pointer for the BASIC program.
basicProgStart	Start of BASIC program area.
varInitOffset	Offset from start of executable to start of variable init area.
varInitStart	Start of variable init area.
keywords	Start of keyword table.
progEnd	End of BASIC program area.
TCver\$	Library file version from executable.
TCtitle\$	TurboCharge copyright message.
jobName\$	Executables job name.

DATAendMarker	Word that marks the end of any DATA in the initialization area.
lineStart	Word that marks 'move.w (a5)+,d6', the start of a BASIC program line.

A6	Address in loaded executable that represents the A6 register.
----	---

keywordOffset	Address that is a base for accessing Keywords. Not a pointer to the keyword list itself.
lineNoExist	1 if compiled program has embedded line numbers 0 if compiled program does not have line numbers.
lineStartPrefix	Prefix code for BASIC program line start, or -1 if no embedded line numbers.
lineCount	Number of program lines decompiled.
lineNo	Line number of line currently de compiling.
pseudoLine (tc)	Address of a reference point when line numbers are suppressed
ch	Channel to send decompiled output to.
er (sc) errCh (tc)	Channel to send Errors & log to.
proc\$	Line numbers of Procedures and Functions, separated by *****
pcount	Number of items in the pipe.
InIF	Number of levels of nested IF's.
IFStack\$	List of 8 character HEX destination addresses of END IF's or ELSE's.
ELSEStack\$	
newLine	0 = Not at the start of a new line.
InPrint	0 = Not in a PRINT/INPUT.
InMultiSel	0 = Not in a multi SElect ON e.g. y=1,2,3,4.
EndSelAdd	0 = Not in a SElect, or address of END SElect.
doPreLoad	1 = Do any preLoad\$.
InProcFun	0 = Not in a Procedure or Function
InSelOn	0 = Not in a SElect, or address of next SElect test, ON REMAINDER, or end of SElect.
SelVar\$	Last used SElect variable.
Indef	0 = Not in any DEFine's
channelOpenCheck	0 = No pending channel operations
key	Holds the current operation prefix code.

## Arrays

nameList()	Array of offsets for keywords at end of code.
arrayElements(20,1)	Holds the number of elements in arrays, 21 entries of 2
DATAarray(30)	DATA statement RESTORE points
codeArray(199)	The code array, 200 entries

### **Turbo Toolkit commands handled in the Keywords section**

ALLOCATION, DEALLOCATE, SEARCH\_MEMORY, PEEK\_F, BASIC\_F, FREE\_MEMORY, BASIC\_B%, BASIC\_W%, BASIC\_L, BASIC\_POINTER, BASIC\_NAME\$, BASIC\_TYPE%, BASIC\_INDEX%, TYPE\_IN, SET\_PRIORITY, COMMAND\_LINE, END\_CMD, CONNECT, CHANNEL\_ID, SET\_CHANNEL, EXECUTE, EXECUTE\_W, EXECUTE\_A, DEFAULT\_DEVICE, LIST\_TASKS, REMOVE\_TASK, SUSPEND\_TASK, RELEASE\_TASK, DATASPACE, CURSOR\_ON, CURSOR\_OFF, EDIT\$, EDITF, EDIT%, SET\_FONT, POSITION, SET\_POSITION, INTEGER\$, FLOAT\$, STRING\$, STRING%, STRINGF, GET%, GETF, GET\$, INPUT\$, DEVICE\_STATUS, DEVICE\_SPACE,

### **Turbo toolkit commands handled by the compiler internally**

PEEK\$, POKE\$, COMPILED, OPTION\_CMD\$, DATA\_AREA (makes an empty line),  
RETRY\_HERE