NAS-DIS 1.0

NAS-DIS is a disassembler, which means that it takes any machine code program and produces an assembler listing from it. This source code, if assembled, is certain to generate the original machine code program again. Disassemblers are very useful in examining machine code programs so that you can work out what they do and make any changes required.

NAS-DIS is a direct descendent of REVAS, and has been written by the same author, David Parkinson. However, it is a vast improvement over the original, and this review can only describe some of its many features. It comes in two versions, either on a tape which can be used to produce a copy of NAS-DIS located at whatever address you choose, or as three EPROMs located at C400H - CFFFH. It is 3K long and runs only under NAS-SYS.

It is most convenient on the Nascom 2 to have a separate memory board for RAM, and to use the eight sockets on the CPU board for 2708 EPROMs. In this case, four can be used for ZEAP 2 (Nascom assembler), and three for NAS-DIS. This makes sense, as NAS-DIS is fully compatible with ZEAP, which means the output of NAS-DIS can be fed into ZEAP without alteration. In fact the assembler output from NAS-DIS can be generated in several ways. It can be simply listed on the screen, or printed, or output to tape, or can even automatically become a ZEAP source file in memory. Printed output can be split into numbered pages with titles, and the tape output is also compatible with ZEAP so that it can be fed directly back into ZEAP.

An option normally selected is the automatic generation of labels. All locations addressed within the program are given labels, making it easy to read the code. Furthermore, a complete cross reference table can be output, showing every address where each label is refered to. This is very valuable when tracing backwards through the logic of a program.

It is also possible to specify that some areas of memory are data areas, so that these are not converted to spurious assembler codes. In this case DEFB instructions appear.

NAS-DIS can, optionally, take account of NAS-SYS restart features, namely SCAL, RCAL and Print String (PRS). With SCAL and RCAL, the next byte is output as a DEFB. With PRS, NAS-DIS generates as many DEFM lines as are needed for the message.

If you don't want to spend any time thinking which options to use, you can simply execute NAS-DIS and specify the start address to disassemble at. This means that any piece of code can be disassembled almost instantly.

Other options in brief:

- Disassemble a program which is not at its normal execution address.
- Change lines per page and page size.
- Display the output from only part of a disassembly of a large program
- NAS-DIS incorporates a subroutine called REVAS (!) which you could call from your own program.
- Good validation of user input when selecting options.

We have tested NAS-DIS and can verify that it disassembles all machine code correctly, with no nasty quirks. It is an excellent piece of software.

NAS-DIS is available from Nascom through your dealers. Ask your dealer about it. If he doesn't know anything about it, nag Nascom Sales Department for prices and ask them why it isn't available.

NAS-DIS

EC400

Options ? (S T Z X L P D R U)

S – source only
T – output on serial port (LED does not light)
Z – Zeap source file. (specify area)
X – cross reference list.
L – labels added.
P – paginated O/P (65 lines/page). (specify title)
D – delay after each line.
R – restricted listing (specify area)
U – Ignore NASCOM special commands e.g. PRS, RCAL.

What on ? (START / END / PROG. COUNT.)

Alternatively just specify EC400 START
and disassembly goes a line at a time.