

IDENTIFICATION

Product Code: DEC-08-COCO-D
Product Name: ODT-8
Date Created: October, 10, 1968
Maintainer: Software Service Group

1 ABSTRACT

ODT (Octal Debugging Technique) is a debugging aid for the PDP-8, which facilitates communication with, and alteration of, the program being run. Communication between operator and program occurs via the Teletype, using defined commands and octal numbers. This version of ODT has been completely revised and replaces both versions of the former ODT-II program.

2 PRELIMINARY REQUIREMENTS

2.1 Equipment

Standard PDP-8 or PDP-5 with basic 4k memory and Teletype.

2.2 Storage

ODT requires 600 (octal) consecutive core locations and one location on page 0 which will be used as an intercom register. It is page relocatable.

3 LOADING OR CALLING PROCEDURE

NOTE: ODT cannot be called as a subroutine.

- a. ODT is normally distributed in binary with the source available on request and is loaded with the Binary Loader.
 1. Place the ODT tape in the reader.
 2. Set 7777 in the SWITCH REGISTER and press LOAD ADDRESS. (If using the high-speed photoelectric reader, put switch 0 down).
 3. Press START.
- b. Load the binary tape of the program to be debugged in the same manner as ODT was loaded. Be sure that the two do not overlap.

4 USING THE PROGRAM OR ROUTINE

4.1 Starting Procedure

- a. The starting address of ODT is the address of the symbol START. For standard library versions the high version starts at 7000 and the low at 1000.
- b. Set the starting address in the SWITCH REGISTER. Press LOAD ADDRESS, and START on the console. ODT will issue a carriage return and line feed to indicate that it is now running and awaiting commands from the keyboard.
- c. To restart ODT without clearing the checksum, set the address of START + 1 (usually 7001 high version, or 1001 low version) into the SWITCH REGISTER and press LOAD ADDRESS and START on the console.

4.2

Control Charactersa. Slash (/) - Open register preceding/

The register examination character / causes the register addressed by the octal number preceding the slash to be opened and its contents typed out in octal. The open register can then be modified by typing the desired octal number and closing the register. Any octal number from 1 to 4 digits in length is a legal input. Typing a fifth digit is an error and will cause the entire modification to be ignored and a question mark to be typed back by ODT. Typing (/) with no preceding argument causes the latest named register to be opened (again). Typing 0/ is interpreted as / with no argument.

Example: 400/6046
 400/6046 2468?
 400/6046 12345?
 /6046

b. Carriage Return (↵) - Close register

If the user has typed a valid octal number, after the content of a register was printed by ODT, typing ↵ causes the binary value of that number to replace the original contents of the opened register and the register to be closed. If nothing has been typed by the user, the register is closed but the content of the register is not changed.

Example: 400/6046↵ Register 400 is unchanged.
 400/6046 2345 ↵ Register 400 is changed to contain 2345.
 /2345 6046 ↵ Replace 6046 in register 400.

Typing another command will also close an opened register.

Example: 400/6046 401/6031 2346 ↵ Register 400 is closed and unchanged and
 400/6046 401/2346 ↵ 401 is opened and changed to 2346.

c. Line Feed (!) - Close register, open next sequential register

The line feed has the same effect as the carriage return, but, in addition, the next sequential register is opened and its contents typed.

Example: 400/6046! Register 400 is closed unchanged and 401
 0401/6031 1234! is opened. User types change, 401 is
 0402/5201 ↵ closed containing 1234 and 402 is opened.

d. Up arrow (↑) - Close register, take contents as memory reference and open same

Up arrow will close an open register just as will carriage return. Further, it will interpret the contents of the register as a memory reference instruction, open the register referenced and type its contents.

Example: 404/3270† 3270 symbolically is "DCA, this page, relative
 0470/0212 0000) location 70," so ODT opens register 470.
 404/3270†
 0470/0000

e. Back Arrow (←) - Close register, open indirectly.

Back arrow will also close the currently open register and then interrupt its contents as the address of the register whose contents it is to type and open for modification.

Example: 365/5760†
 0360/0426 ←
 0426/5201

f. Any Illegal Character

Any character that is neither a valid control character nor an octal digit, or is the fifth octal digit in a series, causes the current line to be ignored and a question mark typed.

Example: 4: ?) } ODT opens no register.
 4U ?) } ODT ignores modification and closes register 406.
 406/4671 67K ?) }
 /4671)

g. xxxxG - Transfer control to user at location xxxx.

Clear the AC then go to the location specified before the G. All indicators and registers will be initialized and the break-trap, if any, will be inserted. Typing G alone is an error but will nevertheless cause a jump to location 0.

h. xxxxB - Set breakpoint at user location xxxx.

Conditions ODT to establish a breakpoint at the location specified before the B. If B is typed alone, ODT removes any previously established breakpoint and restores the original contents of the break location. A breakpoint may be changed to another location, whenever ODT is in control, by simply typing xxxxB where xxxx is the new location. Only one breakpoint may be in effect at one time; therefore, requesting a new breakpoint removes any previously existing one. The previous restriction on placing a breakpoint on a JMS followed by arguments has been removed as of the June 1967 revision. This means ODT can now be more effectively used, especially in debugging programs which utilize floating point. The only restriction in this regard is that a breakpoint may not be set on any of the floating point instructions which appear as arguments of a JMS.