



# DECUS

## PROGRAM LIBRARY

DECUS NO.	12-9
TITLE	SLOWCREF
AUTHOR	John Burness
COMPANY	Digital Equipment Corporation Maynard, Massachusetts
DATE	July 16, 1970
SOURCE LANGUAGE	LAP6-DIAL-MS



## SLOWCREF

DECUS Program Library Write-up

DECUS No. 12-9

### INTRODUCTION

In many situations of assembling, debugging, and modifying programs, a cross-reference listing is an invaluable aid. This listing is particularly useful when a programmer wishes to make changes in a large program or one he does not know well.

A cross-reference is an alphabetical listing of all user-defined symbols with the line numbers at which the symbol was defined and used. Thus, the various places that a tagged location or an equated symbol is used are easily identified.

SLOWCREF is a modified version of the PDP-12 cross-reference program, CREF 12 (DEC-12-FRZA-D), and is used for the special case when a cross-reference of a long system program (e.g. PIP) is needed. CREF 12 is designed to be run on an 8K machine, thereby limiting the size of a program which it can successfully cross-reference. SLOWCREF runs on a 16K machine, thus doubling the size of the program which can be cross-referenced. Because the symbol table crosses field boundaries when doing searches and inserts, SLOWCREF runs from 4 to 8 times slower than CREF 12 on the same program. Therefore, if the user's source is less than about 200 blocks, try to use CREF 12 to cross-reference the program first, rather than SLOWCREF.

It should be noted that the usefulness of a cross-reference is almost directly proportional to the quality of the coding. In particular, the following principles should be used in new coding in order to make the best use of SLOWCREF:

1. Use symbolic references for beta registers, auto-index registers, page zero constants and temporaries, and fixed core locations.
2. Avoid use of large displacements from a tag (e.g., TAD X+12 or JMP Y-6) because references to X and Y, respectively, rather than the locations actually used, will appear in the cross-reference.

### ENVIRONMENT

SLOWCREF (including its header) occupies six blocks in the DIAL file area of any unit. It is loaded into locations 0 through 2377 of field 1 and is entered at 200 in field 1.

### USING SLOWCREF

SLOWCREF is intended for use in conjunction with the LISTAPE pseudo-op of the DIAL-MS Assembler. LISTAPE (with a positive argument) produces a normal assembler listing on a mass storage device, rather than on the console Teletype <sup>(R)</sup> or line printer. The resulting data

---

<sup>R</sup>Teletype is a registered trademark of Teletype Corporation

may then be processed by SLOWCREF to produce a full cross-reference of user-defined symbols and, optionally, the assembly listing.

To use SLOWCREF, insert a LISTAPE n pseudo-op in the program of interest, where n is an expression whose value is between 0 and 17, octal, and is taken as the unit number of a scratch tape (10-17) on which the Assembler listing will be produced.

The user must exercise caution in assigning units for LISTAPE because the listing information is written directly on the unit starting at block zero without regard to anything else that may be on the same unit. Therefore, unit n should be either a scratch tape or a logical disk devoted to scratch work.

### Operating Procedures

1. Set all six sense switches on. Be sure the tape or disk unit assigned by the LISTAPE pseudo-op is ready and WRITE ENABLED.
2. Assemble the program with the LIST command (refer to the LAP6-DIAL Programmer's Reference Manual, DEC-12-SE2D-D).
3. If a line printer is to be used for the cross-reference, it should be ready at this time. When the Editor display returns, load SLOWCREF by typing  $\rightarrow$  LO SLOWCREF,0  $\rightarrow$  .
4. The following message is printed:

UNIT #?

Type an octal number between 0 and 17, followed by carriage return, for the unit on which the listing was written.

5. The next message is:

LISTING [Y OR N] ?

Type Y if the full assembly listing and a cross-reference are desired; N if only the cross-reference is needed.

6. SLOWCREF will then type:

PRINTER [Y OR N] ?

Type Y if the output is to be produced on a line printer (it must be ready before Y is entered), or N if the output is to be to the console Teletype.

Assuming the specified unit is available and ready, SLOWCREF will then read the listing (printing it if requested) and print the cross-reference table.

### Error Conditions

Under some conditions, SLOWCREF will not be able to produce the cross-reference desired.

1. If there were errors in the assembly, SLOWCREF may interpret the program incorrectly, or may even be unable to process it at all. In the latter event, it will print "BAD INPUT" and return to DIAL-MS.
2. If there are a large number of symbols and references, the cross-reference table may overflow. In this case, SLOWCREF prints:

CORE OVERFLOWED AT LINE NO. XXXX

where XXXX is the last line number processed. SLOWCREF prints the cross-reference up to that line, and returns to DIAL-MS. This problem will arise if  $(NSYM*5 + NREF)$  is greater than 8192 where NSYM is the number of user-defined symbols in the program, and NREF is the number of references to user-defined symbols. If there are more than 1638 symbols, this message will occur during the first pass, with XXXX equal to 0000. SLOWCREF will print "BAD INPUT" and return to DIAL-MS.

If a program containing a LISTAPE pseudo-op is to be assembled without getting a cross-reference, use the normal assembly procedure with the sense switches off; this disables the LISTAPE function.

#### INTERNAL DESCRIPTION

After reading the DIAL-MS I/O routines into 7000-7777 of field 1, SLOWCREF requests three parameters from the user: the unit from which to read the listing, if the listing is to be printed, and if a line printer is to be used for the output. SLOWCREF then begins its first pass over the data. The routine MAIN rewinds the input, initializes counters, and calls the subroutines GETLIN and ENDLST to get a line of input and check it for the separator between the listing proper and the symbol table, respectively. This separator is a line with one of two message, "NO ERRORS" or "XXXX ERRORS." If the input runs out (indicated by a word of zero) before the separator is found, the input is invalid and processing is terminated with the message "BAD INPUT." If the separator is found, SYMBUILD takes over and reads the symbol table. Each symbol is stored in stripped ASCII form in field 3 followed by two words of zero. For each symbol added, the table is shifted down towards location 0 of field 2.

When the end of data is found, control passes to PART2, which rewinds the input file. MAJOR calls GETLIN and ENDLST to get successive lines from the file and test for the end of the listing proper. After checking for the separator, MAJOR flows into SCAN, which collects a symbol at NAME, calls NSERCH to search the symbol table for it, and (if it finds the symbol) stores the line number with the symbol in the table. Control returns to SCAN until either carriage return, slash, or "TEXT" is found. Control then returns to MAJOR to get a new line.

When a separator is found, control transfers to DUMP, which prints a header, then reads each symbol from the file. Each symbol and its value are printed, followed by the line number at which the symbol was defined, and then the line numbers at which the references to it were made.

Output is terminated, and control returns to DIAL-MS, when the end of the symbol table is found.

The functions of minor routines are as follows:

BOPUP - Maintains SYSCOUNT to keep track of field  $\emptyset$  utilization and issue the message "CORE OVERFLOWED AT LINE NO. XXXX" when the symbol table overflows. Called by scan when a reference is to be added to the symbol table.

CRLF - Go to a new line on the Teletype or line printer.

GETBLK - Get 8 blocks from the input file. Used by GETLIN when the input buffer is empty, and by MAIN and PART2 to rewind the input and read the first buffer load.

GETC - Obtain the next character from the current line and return it in CHAR and the AC. Used by many routines.

GETLIN - Get a line (a string of characters terminated by zero or line feed) from the input file. If LISTSW is non-zero, print it.

IMOVE - Move a block of data across field boundaries. Used by SYMBUILD to move a new entry into the symbol table.

IPUSH - Move a portion of the symbol table down. During SYMBUILD, the symbol table is built by pushing previous entries down to lower core addresses and entering each new symbol at the end of the list (at the top of field  $\emptyset$ ). During SCAN, references to each symbol are stored following the symbol. IPUSH is called to make room for each entry.

ISNUM - Examine the first four characters of the current line. If not octal digits, return to P+1. Otherwise, convert them to binary, store binary at LINENO, and return to P+2. Called only by MAJOR.

IZERO - Search the symbol table, starting at the location addressed by TEMP, for a word of zero. If none is found, return to P+1. Otherwise, return to P+2 with TEMP pointing to the zero. Called by NSERCH to look for the next symbol and by SCAN to find the end of the references to the current symbol.

LIMIT - Compare the value in the AC to two limits. If the AC is zero, return to address in P+3; if AC is out of limits, return to address in P+4. If AC is within the limits, return to P+5. Used by many routines.

NSERCH - Find the symbol table entry which matches the symbol at NAME. Return to P+1 if no match is found, or to P+2 if successful. Called only by SCAN.

POCTAL - Print the value in the AC, converted to octal characters. Used by BOPUP to print the line number at which overflow occurs and by DUMP to print the cross-reference line numbers.

TTY - Print a character string.