

CIS COBOL ANIMATOR

Operating Guide

Version 1.0

CIS COBOL ANIMATOR Operating Guide: Version 1.0

Copyright © 1982 Micro Focus Ltd

Not to be copied without the consent of Micro Focus.

Micro Focus Limited
58 Acacia Road, St. Johns Wood
London
NW8 6AG

Table of Contents

WELCOME TO ANIMATOR!	vii
RELATED PUBLICATIONS	vii
AUDIENCE	vii
NOTATION IN THIS MANUAL	vii
1. Introduction	1
GENERAL DESCRIPTION	1
GETTING STARTED WITH ANIMATOR	2
ISSUE DISK	2
FIRST STEPS	2
Compilation	2
Running With ANIMATOR	4
CIS COBOL FACILITIES NOT AVAILABLE WITH ANIMATOR	5
ANSI COBOL DEBUG MODULE	5
ALTER STATEMENT	5
COMMAND LINE PARAMETERS	5
2. PREPARING FOR ANIMATION	7
COMPILATION	7
RUNNING	7
3. OPERATING COMMANDS	9
DETAILED COMMAND DESCRIPTIONS	9
SOURCE CODE WINDOW MANIPULATION	9
The S(creen) Command	9
The F(ind) Command	10
EXECUTION AND ANIMATION CONTROL	11
The B(reakpoint) Command	11
The E(xecute) Command	11
The L(evel) Command	12
The P(-C) Program Counter Command	13
EXAMINATION AND AMENDMENT OF DATA	13
The D(isplay) Command	13
The Q(uey) Command	13
The M(onitor) Command	14
USER SCREEN DISPLAY	14
The U(ser) Command	14
A. ANIMATION OPTION COMMAND SUMMARY	15

List of Figures

1.1. System Flowchart	2
-----------------------------	---

WELCOME TO ANIMATOR!

This manual describes your ANIMATOR software which enables you to animate your Micro Focus COBOL application programs with the COBOL source code in front of you on the screen. You can activate parts of the program to suit yourself and at selected speeds and examine data items at break-points or even during running. You can examine and debug your programs using only the source code with no need to refer to details of the object code.

RELATED PUBLICATIONS

Descriptions of the Micro Focus COBOL Products are contained in the following manuals:

CIS COBOL Language Reference Manual

CIS COBOL Operating Guide specific to your Operating System.

AUDIENCE

This manual is intended for COBOL programmers using Micro Focus COBOL development systems.

NOTATION IN THIS MANUAL

Headings are presented in this manual in the following order of importance:

CHAPTER N
Chapter Heading
TITLE

ORDER ONE HEADING
ORDER TWO HEADING
Order Three Heading Text two lines down
Order Four Heading
Order Five Heading: Text on same line

Numbers one (1) to nine (9) are written in text as letters, e.g. one.

Numbers ten (10) upwards are written in text as numbers, e.g. 12.

Chapter 1. Introduction

GENERAL DESCRIPTION

ANIMATOR is a COBOL oriented debugging tool for use with a Micro Focus COBOL product.

The main aim of ANIMATOR is to free the COBOL programmer from the need to be aware of the internal representations of either data or procedural code, so that even a trainee programmer already has the knowledge necessary to debug his programs effectively.

This is achieved by using the screen as a "window" into the source COBOL program and "animating" execution by moving the cursor from statement to statement as execution proceeds. Speed of execution can be varied; the user may also switch off animation thus allowing rapid execution up to the area of interest.

The user can interrupt execution at any point, either by defining break-points or dynamically simply by pressing the space-bar on the keyboard. Whilst execution is suspended the user can easily examine any part of the source code by means of simple commands to refresh the screen display. This means that it is not even necessary to have a printed compilation listing in order to debug a program.

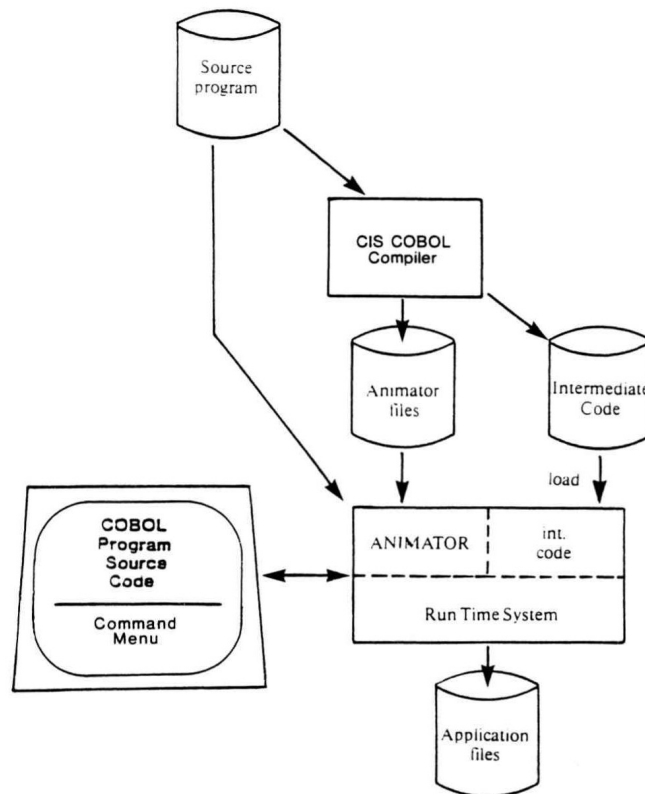
Various other debugging functions are available, invoked by pressing a key. Only the top 20 lines of the screen are used for the display of source code, the bottom area being used to display menus of available commands, some of which invoke subordinate command menus.

Where debugging functions require reference to either data items or procedural statements this is achieved by the user moving the cursor to "point" at the appropriate place in the source code. Alternatively data items can be referenced by actually typing the COBOL data-name.

Where control of ANIMATOR requires more keyboard input than simply pointing with the cursor or pressing one of the displayed command characters, COBOL syntax is used. For instance, replacement of data item values is achieved by typing that value in COBOL literal format (i.e. non-numeric literals are enclosed in quotes).

The facilities provided in ANIMATOR make it much more than simply a COBOL-oriented debugger. It can be a valuable training aid, and also provides the ideal means for a programmer to attain understanding of an unfamiliar program.

Figure 1-1 shows how ANIMATOR fits into the COBOL system.

Figure 1.1. System Flowchart

GETTING STARTED WITH ANIMATOR

ISSUE DISK

Each user is provided with the program that makes up the ANIMATOR debugging tool on an ANIMATOR Issue Disk. This program is supplied in a file called \$ANIM.V45.

FIRST STEPS

You will by now have working CIS COBOL system disks which you created from your CIS COBOL Issue Disk. In order to use ANIMATOR with CIS COBOL you need only copy the file \$ANIM.V45 from your Issue Disk to the CIS COBOL system disk which you are using. Do this now with your CP/M copy software. (Remember you need to register with us as a user of ANIMATOR just as you did with CIS COBOL so fill in your User Registration form *now* if you have not already done so.)

To get off the ground with ANIMATOR this section uses the STOCK1 demonstration program that is described in Chapter 1 of the *CIS COBOL Operating Guide*. Getting started with ANIMATOR basically involves the same processes as were described there. You need to recompile STOCK1 for use with ANIMATOR and, of course, run it. As in the CIS COBOL example, it is assumed that all your files are on one disk in drive A.

Compilation

The compilation command is:

```
A>COBOL STOCK1.CBL ANIM RESEQ<<
```

The compiler will output a title and information just as it does normally. A resequenced listing will be produced in the file STOCK1.LST which you will find handy to print and have next to you, although it is less essential for debugging with ANIMATOR than with conventional debuggers as you will see. We include it here:

```

** CIS COBOL V4.5                                STOCK1.CBL                                PAGE: 0001
**
** OPTIONS SELECTED :
**   COPYLIST RESEQ ANIM
**
000010 IDENTIFICATION DIVISION.                                0118
000020 PROGRAM-ID. STOCK-FILE-SET-UP.                          0118
000030 AUTHOR. MICRO FOCUS LTD.                                0118
000040 ENVIRONMENT DIVISION.                                  0118
000050 CONFIGURATION SECTION.                                0118
000060 SOURCE-COMPUTER. MDS-800.                              0118
000070 OBJECT-COMPUTER. MDS-800.                              0118
000080 SPECIAL-NAMES. CONSOLE IS CRT.                          0118
000090 INPUT-OUTPUT SECTION.                                  0118
000100 FILE-CONTROL.                                          0118
000110     SELECT STOCK-FILE ASSIGN "STOCK.IT"                    0184
000120     ORGANIZATION INDEXED                                0186
000130     ACCESS DYNAMIC                                      0186
000140     RECORD KEY STOCK-CODE.                              0186
000150 DATA DIVISION.                                        01BE
000160 FILE SECTION.                                         01BE
000170 FD STOCK-FILE: RECORD 32.                               01BE
000180 01 STOCK-ITEM.                                         01BE
000190     02 STOCK-CODE PIC X(4).                               01BE
000200     02 PRODUCT-DESC PIC X(24).                             01C2
000210     02 UNIT-SIZE PIC 9(4).                               01DA
000220 WORKING STORAGE SECTION.                                01E0
000230 01 SCREEN-HEADINGS.                                    01E0 00
000240     02 ASK-CODE PIC X(21) VALUE "STOCK CODE      < >".  01E0 00
000250     02 FILLER PIC X(59).                                  01F5 15
000260     02 ASK-DESC PIC X(16) VALUE "DESCRIPTION    <".    0230 50
000270     02 SI-DESC PIC X(25) VALUE "                >".    0240 60
000280     02 FILLER PIC(39).                                      0259 79
000290     02 ASK-SIZE PIC X(21) VALUE "UNIT SIZE      < >".  0280 A0
000300 01 ENTER-IT REDEFINES SCREEN-HEADINGS.                  01E0 00
000310     02 FILLER PIC X(16).                                    01E0 00
000320     02 CRT-STOCK-CODE PIC X(4).                             01F0 10
000330     02 FILLER PIC X(76).                                  01F4 14
000340     02 CRT-PROD-DESC PIC X(24).                             0240 60
000350     02 FILLER PIC X(56).                                    0258 78
000360     02 CRT-UNIT-SIZE PIC 9(4).                             0290 B0
000370     02 FILLER PIC X.                                       0294 B4
000380 PROCEDURE DIVISION.                                      0000
000390 SR1.                                                       001C 00
000400     DISPLAY SPACE.                                         001D
000410     OPEN I-O STOCK-FILE.                                   0020
000420     DISPLAY SCREEN-HEADINGS.                              0024
000430 NORMAL-INPUT.                                           0038 00
000440     MOVE SPACE TO ENTER-IT.                                0039
000450     DISPLAY ENTER-IT.                                      003F
000460 CORRECT-ERROR.                                          0056 00
000470     ACCEPT ENTER-IT.                                       0057
000480     IF CRT-STOCK-CODE = SPACE GO TO END-IT.              006E

```

```

000490     IF CRT-UNIT-SIZE NOT NUMERIC GO TO CORRECT-ERROR.          0078
000500     MOVE CRT-PROD-DESC TO PRODUCT-DESC.                        0081
000510     MOVE CRT-UNIT-SIZE TO UNIT-SIZE.                           0087
000520     MOVE CRT-STOCK-CODE TO STOCK-CODE.                         008F
000530     WRITE STOCK-ITEM INVALID GO TO CORRECT-ERROR.            0095
000540     GO TO NORMAL-INPUT.                                        00A1
000550 END-IT.                                                       00A4 00
000560     CLOSE STOCK-FILE.                                          00A5
000570     DISPLAY SPACE.                                             00A9
000580     DISPLAY "END OF PROGRAM".                                  00AC
000590     STOP RUN.                                                 00BD
** CIS COBOL V4.5 REVISION 0                                         URN AA/0000/AA
** COMPILER COPYRIGHT (C) 1978,1982 MICRO FOCUS LTD
** ERRORS=00000 DATA=00768 CODE=00256 DICT=00362:21117/21479 GSA FLAGS= OFF

```

Running With ANIMATOR

Enter the normal CIS COBOL run command:

```
A>RUNA +A STOCK1. INT<<
```

The run-time system software displays its header line:

```
**CIS RTS V4.5 COPYRIGHT (C) 1978, 1982 MICRO FOCUS LTD. URN XX/nnnn/XX
```

This is replaced by the ANIMATOR header lines

```

CIS COBOL ANIMATOR V1.0

COPYRIGHT (C) 1981, 1982 MICRO FOCUS LTD

URN XX/nnnn/XX

```

This is in turn replaced by the COBOL source listing of the program on the screen, starting at the first line of the Procedure Division with the cursor positioned under the first verb. Available commands are displayed on a line below this listing display.

You are now ready to "animate" the STOCK1 program. To give you a "feel" for using ANIMATOR, some arbitrary instructions follow. You can, of course, carry out any of the commands given in Chapter 3. Chapter 3 describes all the commands and error lines in details.

Enter F for the Find command. The command line goes blank. Enter "CRT-STOCK-CODE" (the delimiters are necessary) and press RETURN. ANIMATOR will find the first reference to this data-name (line 000480).

Enter M to Monitor this data-item. The Monitor options are displayed. Press S for Set. A line of # symbols flashes on the screen. This denotes an error. The cursor is positioned at the end of the data-item. It must be at the first character to set monitoring. Move it there by use of the "character-back" key on your keyboard, and enter M and then S again. Nothing happens yet because the data-item is blank. As soon as you fill it, the contents will be displayed under the command line.

Enter B to set a Breakpoint (i.e. where the program will stop whenever it passes that point). The Breakpoint options are displayed. This time we realize we are in error. You can only set breakpoints at statement verbs. We therefore press the space bar to reset to the command line. Move the cursor down to the MOVE verb on line 00500 and position it under the M.

Enter B and then S to set the breakpoint here.

We now enter E for Execute and the Execute options are displayed. Press G for Go and execute starts at the default speed (4). Enter 1 to select the slowest execution speed so we can see what is happening.

The cursor now runs through the COBOL code as it executes, bringing the COBOL program to life or "animating" it. You can halt animation at any time by pressing the space bar and restart by pressing G again. Note that you can enter G without the E(xecute) command.

The data entry screen is displayed and you enter values as you did when getting started with CIS COBOL.

After accepting your data the program stops at your breakpoint as set (line 00500). Notice that the value you entered as STOCK CODE is now displayed under the command line because you asked to monitor its content.

Enter G to go again and then select speed 9. Notice the difference in animation speed. Enter a new record into your data entry screen and the program stops at the breakpoint again with the new value showing in the monitored data-item under the command line.

Enter Z (for Zoom). Note that this is another E(xecute) command that you can enter directly. Animation stops and the program reverts to normal execution. If you enter a blank stock code it terminates.

You may now revert to the **RUN +A** command and repeat it, but this time using the commands from Chapter 3 and experimenting.

Appendix A contains a command summary and the *CIS COBOL Pocket Guide* contains a more concise summary.

CIS COBOL FACILITIES NOT AVAILABLE WITH ANIMATOR

This section describes constraints on the use of a CIS COBOL program with ANIMATOR.

ANSI COBOL DEBUG MODULE

The COBOL Debug module provides facilities for monitoring procedures by means of code activated by an object time switch. Object time activation of this code is inhibited by the use of ANIMATOR, therefore this object time switch should not be set.

ALTER STATEMENT

ANIMATOR makes use of a paging mechanism so that the animation of large programs is not prevented by the presence of the ANIMATOR module.

This has the effect that the state of **ALTERed GOTO** statements cannot be predicted with any certainty. Therefore care must be exercised on the use of ANIMATOR with programs containing ALTER statements.

However, note that ANIMATOR provides facilities enabling the effect of such control transfers to be seen, and to reset the point of program execution if required.

COMMAND LINE PARAMETERS

CIS COBOL provides the facility to read parameters entered on the command line by means of the first ANSI-format **ACCEPT** statement executed.

This facility is inhibited when animating; such parameters must be entered at the time the **ACCEPT** statement is executed.

Chapter 2. PREPARING FOR ANIMATION

COMPILATION

The steps involved in compiling a program are described in detail in the *CIS COBOL Operating Guide*.

Note that if it may be required to animate the program the compiler directive ANIM must be specified.

Two other compiler directives can be particularly useful; these are COPYLIST and RESEQ. Use of these directives will ensure that the compilation listing matches the ANIMATOR screen display.

The use of the ANIM directive causes certain files required for animation to be produced in addition to the normal intermediate code file(s). These will have the extensions .DDC, .SDB and .SCP. The main source file must have the extension .CBL.

Note that the program can be run without animation and will behave exactly as if the ANIM directive had not been used for compilation.

RUNNING

The invocation of ANIMATOR to run a program is described in the *CIS COBOL Operating Guide*.

The "+A" directive specifies that animation is required. The ANIMATOR module itself must be present on the default drive. The special ANIMATOR files (.DDC, .SDB, .SCP), along with the main source file (which must have the extension .CBL), should be present on the same device as the intermediate code if animation of the particular program is required. If any of these files is missing the program is executed normally - this allows particular programs within a hierarchy to be selected for animation.

Any library (COPY) files should also be present, on the device specified by the COPY statement or, if none is specified, on the same device as the intermediate code.

Chapter 3. OPERATING COMMANDS

When a program is loaded for animation, as described in Chapter 2, the first screenful of procedural code is displayed with the cursor positioned on the first executable statement. Execution will not commence until initiated by a command as described in this Chapter.

At the bottom of the screen is a delimiting line of hyphens followed by the main command menu:-

```
B(rk-pnts) D(isp) E(xec) F(ind) L(evel) M(on) P(-c) Q(uey) S(creen) U(ser)
```

Commands are entered by simply pressing the appropriate key. If an invalid entry is made at any time ANIMATOR indicates this by briefly replacing the prompt line with a line of hash (#) signs and "bleeping".

Several of the main commands merely serve to invoke subordinate menus of more detailed commands. For convenience some of the more commonly used of these subordinate commands can also be entered directly whilst the main prompt line is displayed (see detailed descriptions). After execution of a subordinate command the main prompt line is redisplayed.

Some of the commands require that the cursor is first moved to point to the appropriate position in the displayed source code. The main prompt line shown above is displayed whenever execution is suspended. Whilst this prompt line is displayed the cursor can be moved over the screen using the normal cursor control keys. For details of these (and how to change them) refer to the *CIS COBOL Operating Guide*.

Note that with ANIMATOR the back field and forward field function keys are used to move the cursor up and down within a screen column. In addition RETURN will move the cursor to the start of the next line.

DETAILED COMMAND DESCRIPTIONS

The main commands fall into 4 broad categories:

- Source code "window" manipulation (S,F)
- Execution and animation control (B,E,L,P)
- Examination and amendment of data (D,Q,M)
- User screen display (U)

These commands may only be entered when the main prompt line is displayed.

SOURCE CODE WINDOW MANIPULATION

ANIMATOR uses the screen as a window into the source code text. These commands allow you to reposition the source code window to any point within the source COBOL program. Note that this in itself does not affect point at which execution will be resumed.

ANIMATOR automatically displays resequenced line numbers against the source text. These will be the same as those appearing on the compilation listing if the directives RESEQ and COPYLIST (if appropriate) were specified to the compiler.

The S(screen) Command

Press S and the following subordinate menu is displayed:

```
SCREEN - N(ext) P(revious) T(op) E(nd) V(iew) H(alf) F(ull) =/+/-
```

These commands reposition the window to display a different part of the source text.

Press the appropriate key where:

- N displays Next screen from source text. Note that this overlaps the current screen by two lines.
- P displays Previous screen from source text.
- T displays screen at Top of source text.
- E displays screen at End of source text.
- V repositions window so that the source line indicated by the cursor is on the third line. Note: the cursor must be positioned before pressing S.
- H Splits screen in Half (ie into two windows) with a dividing line of hyphens. The lower window is positioned to show the top of source text. Note: Subsequent screen commands operate on the window in which the cursor is positioned.
- F restores Full screen display (single window).
- =*n* repositions the window such that the *n*th source line is aligned at the third screen line. Note: *n* is equivalent to the displayed line number omitting the trailing zero.
- +*n* moves the window forward *n* lines.
- n* moves the window back *n* lines.

Note: =, +, - all position the cursor for entry of a numeric quantity followed by RETURN.

There is one special case of the screen command. If there is a split screen display and the cursor is positioned on the dividing line of hyphens, then when S is pressed the following subordinate menu is displayed:

```
SCREEN DIVIDER - U(p) D(own)
```

These commands allow the relative size of the two windows to be altered.

- U moves the screen divider Up one line.
- D moves the screen divider Down one line.

The F(ind) Command

This command instructs ANIMATOR to search forward from the current cursor position through the source text for a specified string of characters. If found the screen window is positioned with the line containing this string as the third screen line and the cursor is positioned following the string. If not found the main screen display remains unchanged, but ANIMATOR indicates the failure by bleeping and restoring the main prompt line.

Press F and the cursor is positioned for entry of:

Either

"string" followed by RETURN

or

"string"M followed by RETURN

where:

- any character not forming part of the string can be used in place of "."
- string is any sequence of characters (including spaces); it need not be a complete word.
- the optional M instructs ANIMATOR to only search the main source file and not any library (COPY) files.

Note

ANIMATOR only examines columns 7 - 72 of the source text, the displayed line numbers are ignored.

EXECUTION AND ANIMATION CONTROL

These commands allow initiation and control of program execution, also control of the degree of animation.

The B(reakpoint) Command

Press B and the following subordinate menu is displayed:

```
BREAK-POINTS - S(et) U(nset) C(ancel) eX(amine)
```

These commands allow break-points to be set at which execution will halt automatically. Some of these commands require the cursor to be positioned to point to the relevant COBOL statement. The cursor must be positioned on the first character of the COBOL verb, prior to pressing B.

Up to 4 break-points may be set concurrently.

Press the appropriate key, where:

- S Sets a break-point at the statement pointed to by the cursor.
- U Unsets the break-point pointed to by the cursor.
- C Cancels all break-points currently set.
- X eXamines break-points by repositioning the window within the source code and positioning the cursor at the relevant statement. Successive calls on this function will move the cursor to point to each break-point in turn.

The E(xecute) Command

Press E and the following subordinate menu is displayed:

```
EXECUTE - X(single step) sK(ip) I(till next If) G(o) Z(oom) S(top run)
```

These commands initiate execution, with or without animation, in a variety of ways.

Note

Any of these commands except for S(top run) may also be entered directly against the main prompt line without the preceding E.

Press the appropriate key, where:

- X executes a single COBOL statement and moves the cursor to the next statement.

K skips a single COBOL statement, without executing it, and moves the cursor to the next statement.

Note: If the final statement of a **PERFORM**ed paragraph is skipped, control does not exit from the **PERFORM** but passes to the next statement in the source code.

I Executes without animation up to the next **IF** statement, at which point execution halts and the cursor is repositioned at this **IF** statement.

G Initiates animated execution. As each statement is executed the cursor is moved to the next statement in the source code. The speed of animated execution can be varied by typing a digit from 1 to 9 (1 = slowest, 9 = fastest). The speed may also be entered before initiating execution. Execution proceeds until halted as described below.

Z Initiates execution without animation (Zooms). Upon reaching the first **DISPLAY UPON CRT** or **ACCEPT FROM CRT** the user screen is displayed, replacing the source code, and remains on the screen until execution is halted as described below.

S Stops execution after displaying the current user screen.

After initiation by one of the above commands, execution proceeds as described above, unless it is halted in one of the following ways:

1. If the space-bar is pressed execution is immediately halted.
2. If a previously set break-point is reached execution is automatically halted.
3. If execution is about to "drop through" the end of the program the following prompt is displayed:

```
WARNING - Next instruction is implied STOP RUN - S(top run) C(ontinue)
```

Press any key to restore the prompt line and regain control. If execution is restarted without first resetting the program counter the program will terminate; see the P(-C) command.

4. If a run-time error occurs the following prompt is displayed:

```
RTS ERROR: nnn - S(top) C(ontinue)
```

Press S to stop, or C to regain control.

Error numbers are detailed in the *CIS COBOL Operating Guide*.

The L(evel) Command

Animation normally traces execution into any level of nested **PERFORM**. This command allows a "threshold" level to be set at any level of nesting, such that any **PERFORM**'s subordinate to this level are treated as a single statement for animation purposes, i.e. the cursor will not be moved into **PERFORM**ed procedures below the threshold level.

Press L and the following subordinate menu is displayed:-

```
PERFORM LEVEL - S(et) U(nset) E(xit)
```

Press the appropriate key, where:

- S sets the threshold level at the current level.
- U unsets the threshold level, restoring animation at all levels.
- E completes execution of the current **PERFORM** without animation, repositioning the cursor to the statement following the **PERFORM**, and then sets the threshold level at this point.

The P(-C) Program Counter Command

This command provides facilities to ascertain the point at which execution will start (or resume), or to change this point. To change the restart point it is first necessary to place the cursor at the required position.

Press P and the following subordinate menu is displayed:

```
PROGRAM COUNTER - W( here ) R( eset )
```

Press the appropriate key, where:

W repositions the screen window as necessary and positions the cursor at the next statement to be executed. This is useful as a check after use of the source screen manipulation commands, but note that it is not necessary since ANIMATOR will resume execution at the correct position unless the following command is used.

This command may also be entered against the main prompt line (without the preceding P).

R resets the execution start point (program counter) to the current cursor position. Before pressing P the cursor should be positioned on the first character of an executable statement (ie a COBOL verb).

EXAMINATION AND AMENDMENT OF DATA

These commands provide facilities for the examination and amendment of the contents of specified data items. Selected data items are indicated by either specifying the dataname or pointing with the cursor.

The D(isplay) Command

This command allows display/amendment of a data item referenced by typing the name.

Press D and the cursor is positioned for entry of the required data-name followed by RETURN.

The value of the specified data item is displayed with appropriate conversion for usages COMP and COMP-3 (see the *CIS COBOL Language Reference Manual*). For alphanumeric or group items, non-ASCII characters are displayed as ^.

For signed numeric data items a leading sign is displayed.

Following display of the data item the cursor is positioned to the next line for entry of a replacement value if required.

If no replacement is required simply press RETURN, otherwise enter the new value in COBOL literal format (ie alphanumeric literals must be in quotes). Replacement is performed in accordance with the rules for the COBOL **MOVE** statement.

Note

Only up to 80 characters will be displayed. The attempted replacement of a longer data item results in padding with spaces in accordance with the rules for a **MOVE** statement (See the *CIS COBOL Language Reference Manual*).

The Q(uey) Command

This command allows display/amendment of a data item referenced by pointing with the cursor.

Position the cursor to any occurrence of the data-name within the source code, then:

Press Q

The data item is displayed and may be amended as described for the D(isplay) command.

The M(onitor) Command

This command enables automatically repeated display of a single specified data item (without amendment).

Press M and the following subordinate menu is displayed.

MONITOR - S(et) U(nset) N(ame)

Press the appropriate key, where:

S sets the monitor on the data item referenced by the cursor. The cursor must be moved to point to any occurrence of the data-name before pressing M.

U unsets the monitor.

N positions the cursor for entry of the required data-name followed by RETURN.

The monitored item is redisplayed after each animation "step" - eg after each statement if in "GO" mode.

USER SCREEN DISPLAY

During animation any ANSI format ACCEPT/DISPLAY statements are diverted to the bottom of the screen so that they do not interfere with ANIMATOR's use of the screen.

This approach is not possible for the **ACCEPT FROM CRT** and **DISPLAY UPON CRT** extensions for full screen interaction. Therefore all such ACCEPT/DISPLAY screen data is buffered internally.

This "user screen" automatically replaces the source code window display when an ACCEPT is executed, so that the user can interact with the screen in the normal way.

Additionally the user screen can be examined by means of the U(ser) command.

The U(ser) Command

Press U and the user screen is displayed, replacing the source code window display. If the Z key is pressed at this point, execution without animation (Zoom) is initiated, and the user screen remains on display. Otherwise, if any other key is pressed, the display reverts to the source code.

Appendix A. ANIMATION OPTION COMMAND SUMMARY

This Appendix is a summary in alphabetical order (rather than logical order) of the commands that the user can select from the CRT menu that is displayed when the Animation Option is invoked in the Run command. A brief summary of each command is given. A summary is also contained in the CIS COBOL Pocket Guide.

- B The Breakpoint command allows breakpoints to be set by the user at which execution will halt automatically.

The Breakpoint command offers a menu of four options:

- S Sets a break-point at the statement pointed to by the cursor.
 - U Unsets the break-point pointed to by the cursor.
 - C Cancels all break-points currently set.
 - X eXamines break-points by repositioning the window within the source code and positioning the cursor at the relevant statement. Successive calls on this function will move the cursor to point to each break-point in turn.
- D The Display command enables display and/or amendment of the named data-item.
 - E The Execute command is used to specify the way in which the user requires execution of the program. On entry, the command displays an option menu as follows:
 - X EXecutes a single COBOL statement and moves the cursor to the next statement.
 - K SKips a single COBOL statement, without executing it, and moves the cursor to the next statement.
 - I Executes (without Animation) to the next If statement, halts and positions the cursor at this statement.
 - G Initiates execution (GO) with Animation. As each statement is executed (at a specified speed) the cursor is moved to the next statement in the source code. Speed of execution can be set before entering G or altered after entering G by entry of a numeric character 1 thru 9, where 1 is slowest and 9 is fastest.
 - Z Initiates execution without Animation activated (Zooms)
 - S Stops execution after displaying the current user screen.
 - F The Find command searches from the current cursor position through the source text for a specified string of characters.
 - L The Level command allows a "threshold" level to be set at any level of nested PERFORM such that any PERFORM statements at this level are treated as a single statement for Animation purposes, i.e., the cursor will not be moved into PERFORMed procedures below threshold level.
 - S Sets the threshold at the current level.
 - U Unsets the threshold level and restores Animation at all levels.
 - E Completes the current **PERFORM** without further Animation, then sets the threshold at the level of the statement immediately succeeding the current **PERFORM** statement.

- M The Monitor command enables automatic repeated display of a single specified data-item (without amendment) during program execution.
- S Sets the Monitor on the data-item at the current cursor position.
- U Unsets the Monitor.
- N Positions the cursor for operator entry of the Name of the data-item to be Monitored.
- P The program-counter command provides facilities to ascertain the point at which execution will start (or resume), or to alter this point. On entry, the P command displays an option menu as follows:
- W Where - repositions the cursor at the next statement to be executed.
- R Resets the execution start point to the current cursor position.
- Q The Query command allows display and/or amendment of the data-item pointed at by the cursor when Q is entered.
- S The Screen command repositions the screen window to display a different part of the source text as follows:
- N Displays Next screen from source text.
- P Displays Previous screen from source text.
- T Displays screen at Top of source text.
- E Displays screen at End of source text.
- V Repositions window so that the source line indicated by the cursor is on the third line. Note: the cursor must be positioned before pressing S.
- H Splits screen in Half (ie into two windows) with a dividing line of hyphens. The lower window is positioned to show the top of source text. Note: Subsequent screen commands operate on the window in which the cursor is positioned.
- F Restores Full screen display (single window).
- =*n* Repositions the window such that the *n*th source line is aligned at the third screen line.
- +*n* Moves the window forward *n* lines.
- n* Moves the window back *n* lines.

Note: =, +, - all position the cursor for entry of a numeric quantity followed by RETURN.

There is one special case of the screen command. If there is a split screen display and the cursor is positioned on the dividing line of hyphens, then when S is pressed the following subordinate menu is displayed:

SCREEN DIVIDER - U(p) D(own)

These commands allow the relative size of the two windows to be altered.

U moves the screen divider Up one line.

D moves the screen divider Down one line.

- U The User command displays the current user screen replacing the source code window display until any key is pressed.

Colophon

This book was reconstructed into DocBook format from a scanned PDF found on the Internet. The PDF file already had OCR performed and the text was embedded in the file.

Source version: 1.0

