
LEVEL II COBOL Animator
Operating Guide
UNIX Operating System Implementation
Version 2.0

Document Release 1.2
June 24, 1983

Date of Printing: June 29, 1983

This document has been prepared for the exclusive use of the client. None of the material herein may be reproduced, copied or distributed in any manner, including electronically, without prior written approval of The Santa Cruz Operation, Inc.

Copyright (c) 1982, 1983

The Santa Cruz Operation, Inc.

TABLE OF CONTENTS

NOTICE	i
PREFACE	ii
1. INTRODUCTION	1-1
General Description	1-1
2. PREPARING FOR ANIMATION	2-1
Compilation	2-1
Running Programs with ANIMATOR	2-2
3. OPERATING COMMANDS	3-1
Main Commands	3-2
Source Code Window Manipulation	3-3
Execution and Animation Control	3-6
Examination and Amendment of Data	3-14
APPENDIX A.	
ANIMATOR COMMAND SUMMARY	A-1

NOTICE

COBOL is an industry language and is not the property of any company or group of companies, or of any organization or group of organizations.

No warranty, expressed or implied, is made by any contributor or by the CODASYL Programming Language Committee as to the accuracy and functioning of the programming system and language. Moreover, no responsibility is assumed by any contributor, or by the committee, in connection herewith.

The authors and copyright holders of the copyrighted material used herein:

FLOW-MATIC (Trademark for Sperry Rand Corporation) Programming for the Univac I and II, Data Automation Systems copyrighted 1958, 1959, by Sperry Rand Corporation; IBM Commercial Translator Form No. F28-8013, copyrighted 1959 by IBM; FACT, DS127A5260-2760, copyrighted 1960 by Minneapolis-Honeywell.

have specifically authorized the use of this material in whole or in part, in the COBOL specifications. Such authorization extends to the reproduction and use of COBOL specifications in programming manuals or similar publications.

Note that the following are registered trademarks:

LEVEL II (L/II) COBOL is a trademark of Micro Focus Ltd.
ANIMATOR is a trademark of Micro Focus Ltd.
UNIX is a trademark of Bell Laboratories.

PREFACE

ANIMATOR enables you to step through your Micro Focus COBOL application programs with the COBOL source code in front of you on the screen. You can select parts of the program to activate and can specify the speed of activation. You can also examine data items at break-points or even during program execution. You can examine and debug your programs using only the source code, with no need to refer to details of the object code.

Audience

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

Notation

Throughout this manual, the following notation is used to describe the format of data input or output:

1. When material is enclosed in square brackets [], it is an indication that the material is an option which may be included or omitted as required.
2. The symbol << after a CRT entry or command format in this manual indicates that the CR (carriage return) or equivalent data input terminator key must be pressed to enter the command.

PREFACE

Related Publications

For details of the L/II COBOL Language, refer to the document:

L/II COBOL Language Reference Manual

For details of the UNIX Operating System, error messages and file structures, refer to the UNIX Programmer's Manual.

CHAPTER 1. INTRODUCTION

A. General Description

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

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. You can vary the speed of execution or switch off animation, thus allowing rapid execution up to the area of interest.

You can also interrupt execution at any point, either by defining break-points, or dynamically by pressing the interrupt key and then space-bar. While execution is suspended, you can easily examine any part of the source code by means of simple commands to refresh the screen display. This means that you do not need to have a printed compilation listing in order to debug a program.

CHAPTER 1. INTRODUCTION

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

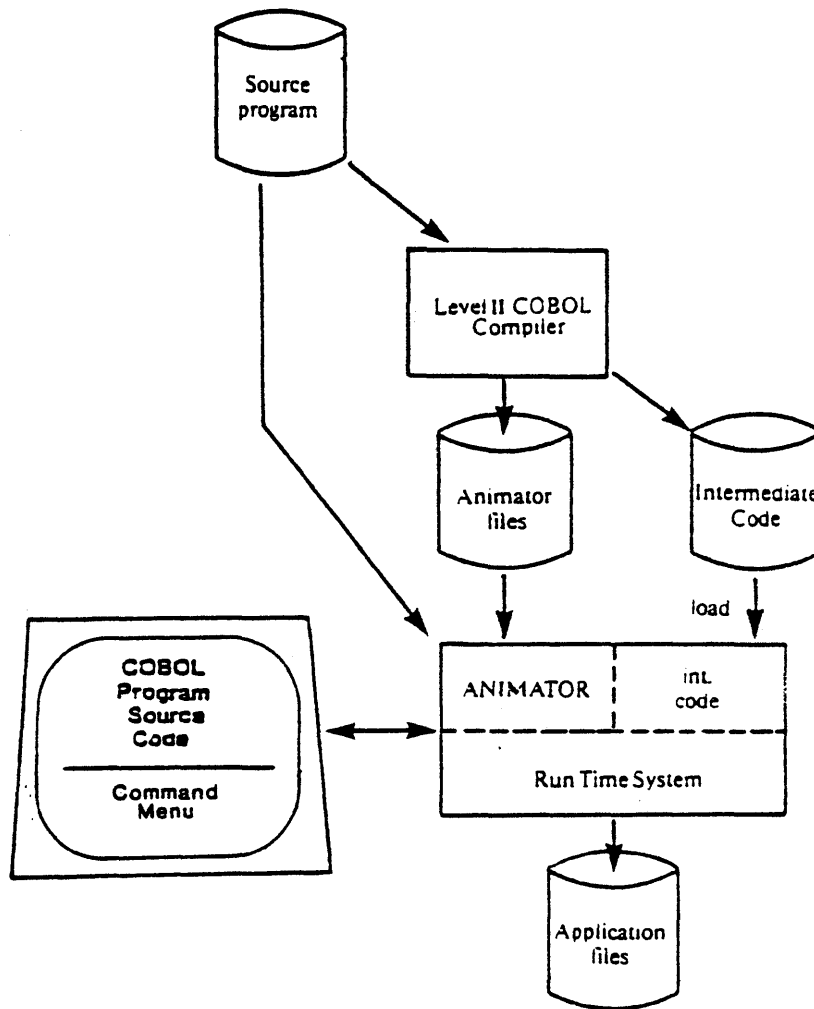
When a debugging function you are using requires reference to either data items or procedural statements, you can move the cursor to "point" at the appropriate place in the source code. Alternatively, you can reference data items by actually typing the COBOL data-name.

When control of ANIMATOR requires more keyboard input than simply pointing with the cursor or pressing one of the displayed command characters, you can use COBOL syntax. For instance, you can replace data item values by typing that value in COBOL literal format (i.e. non-numeric literals are enclosed in quotes). It is even possible to type in a complete COBOL source statement for immediate execution.

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

CHAPTER 1. INTRODUCTION

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



CHAPTER 2. PREPARING FOR ANIMATION

A. Compilation

In order to use ANIMATOR with a LEVEL II COBOL program, you must first compile the program for animation. In the UNIX environment, this means using the command "canim" to compile the program. The command format is:

```
canim prog.cbl [ options ] [ directives ]<<
```

or

```
canim [ options ] prog.cbl [ directives ]<<
```

The set of available options and directives is the same as those for the "cobol" command. (See your LEVEL II Operating Guide for more information) The "canim" command provides some additional necessary information for the compiler, including the ANIM directive.

Compiling using "canim" produces certain files required for animation in addition to the normal intermediate code file(s). These will have the extensions .ANM, .ACP, .DOO and, if the program is segmented, .Dnn where "nn" represents the segment number. These files, the main source file, and any library (COPY)

CHAPTER 2. PREPARING FOR ANIMATION

files, must be present during an animated run.

Two compiler directives can be particularly useful when compiling for animation; these are COPYLIST and RESEQ. Using these directives ensures that the compilation listing matches the ANIMATOR screen display.

Note that the program can be run without animation and will behave exactly as if it had been compiled with the "cobol" command.

B. Running Programs with ANIMATOR

The "animate" command is used to run a LEVEL II COBOL program with animation. The command format is:

```
animate prog<<
```

or

```
animate prog.INT<<
```

If the the provided file name does not have an extension (does not contain a '.' character) the extension '.INT' will be appended. The "animate" command is very similar to the "canim" command; it provides the RNIM directive which causes the execution to be animated. If a particular program has not been compiled with "canim" (but instead with "cobol") then execution will continue without animation.

CHAPTER 2. PREPARING FOR ANIMATION

Note that the following particular COBOL constructions cannot be animated:

COPY ... REPLACING ...

COPY (on several lines)

COPY (followed by further COBOL statements on the same line)

CHAPTER 3. OPERATING COMMANDS

When you run a program with animation, the first screen full of procedural code is displayed with the cursor positioned on the first executable statement. Execution will not begin until you give a command as described in this Chapter. At the bottom of the screen is a delimiting line of hyphens followed by the first part of 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) ?
```

You can review the remainder of the main command menu by pressing "?". It is displayed as follows:-

```
C(ompile) N(ame) unT(il) lOcA(te) ?
```

If you enter "?" again, the first part of the main command menu shown above is redisplayed.

You can enter commands by pressing the appropriate key. If you make an invalid entry at any time, ANIMATOR indicates this by briefly replacing the prompt line with a line of pound (#) signs and beeping.

CHAPTER 3. OPERATING COMMANDS

Several of the main commands serve only to display menus of more commands. You can enter some of the more commonly used of these subordinate commands even though the main prompt line is displayed. Commands that can be entered in this way are described later in this chapter. After execution of a subordinate command, the main prompt line is redisplayed.

Some of the commands require you to first move the cursor to the appropriate position in the displayed source code. The main prompt line (shown above) is displayed whenever execution is suspended. While this prompt line is displayed, you can move the cursor over the screen using the normal cursor control keys. For details on the use of the cursor control keys, refer to the L/II COBOL Operating Guide.

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

A. Main Commands

The main commands fall into four broad categories:

- Source code "window" manipulation (S,F,O,A)
- Execution and animation control (B,T,E,L,P,C,N)

CHAPTER 3. OPERATING COMMANDS

- 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.

B. Source Code Window Manipulation

ANIMATOR uses the screen as a window into the source code text. The commands S, F, O, and A allow you to reposition the source code window to any point within the COBOL source program. Note that this in itself does not affect the point at which execution will resume.

ANIMATOR automatically displays resequenced line numbers within the source text. These line numbers will be the same as those appearing on the compilation listing if you specified the directives RESEQ and COPYLIST.

1. 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.

CHAPTER 3. OPERATING COMMANDS

Press the appropriate key:

- 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: you must position the cursor before you press S.
- H Splits screen in Half (i.e. 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 nth source line is aligned at the third screen line. Note: n is equivalent to the displayed line number, the trailing zero omitted.
- +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 you press S, the following subordinate menu is displayed:

SCREEN DIVIDER - U(p) D(own)

CHAPTER 3. OPERATING COMMANDS

These commands allow you to alter the relative size of the two windows.

U moves the screen divider Up one line.

D moves the screen divider Down one line.

2. The F(ind) Command

This command tells ANIMATOR to search forward from the current cursor position through the source text for a specified string of characters. If the string is found, the screen window is positioned with the line containing the string. If it is not found, the terminal beeps to indicate the failure, and the main prompt line is restored.

Press F and the cursor is positioned for entry of either:

```
/string/<<
```

or

```
/string/M<<
```

where:

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

CHAPTER 3. OPERATING COMMANDS

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

3. The lOcA(te) Commands

The locate commands (O and A) allow you to locate the declaration of a data-name or procedure-name. You can use these commands in the following ways:

1. Position the cursor to point to any occurrence of the name, then press O, or,
2. Press A, type the name, and then press RETURN.

The screen window is repositioned in the source text with the cursor pointing to the declaration of the specified name.

C. Execution and Animation Control

These commands allow you to initiate and control program execution, and also to control the degree of animation.

1. The B(reakpoint) Command

Press B and the following subordinate menu is displayed:

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

These commands allow you to set break-points which will halt execution automatically. A break-point may have an associated

CHAPTER 3. OPERATING COMMANDS

condition. Some of the break-point commands require you to position the cursor at the relevant COBOL statement. You must position the cursor 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:

- 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.
- I Sets a break-point with an associated condition at the statement pointed to by the cursor. ANIMATOR repositions the cursor for entry of the required conditional expression in COBOL format which must be followed by RETURN. Subsequent examination of break-points (see X above) will both point to this break-point and also display the associated condition. When execution reaches this break-point it will only halt if the condition is true.

2. unT(il) Command

This command allows you to specify a conditional break-point, i.e. execution halts if ever a specified condition becomes true during execution. When execution halts, the condition is automatically "switched off".

CHAPTER 3. OPERATING COMMANDS

Note that this facility significantly degrades the speed of execution.

Press T and the following subordinate menu is displayed:-

UNTIL CONDITION - S(et) U(nset) eX(amine)

Press the appropriate key, where:-

S positions the cursor for entry of the required conditional expression in COBOL format.

U cancels the previously set condition.

X displays the previously set conditional expression.

3. 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:

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

CHAPTER 3. OPERATING COMMANDS

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

NOTE: If the final statement of a PERFORMed 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 pressing the interrupt key for your terminal and then 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.

If you specify execution, you may halt it in one of the following ways:

1. If you press the interrupt key (DELETE, RUBOUT, or CTRL-C) once and then press the space-bar, execution halts immediately.

IMPORTANT: Even though the screen display specifies pressing only space-bar to halt, you MUST press the interrupt key first. This sends a signal to ANIMATOR that tells it to read from the keyboard. The space-bar is then seen as the command to halt execution.

2. If a set break-point is reached, execution is automatically halted.

CHAPTER 3. OPERATING COMMANDS

3. If the condition specified by means of the unT(ii) command becomes satisfied, execution is halted.
4. If a STOP RUN statement is reached the following prompt is displayed:

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

Press S to stop, or C to regain control.

5. 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 L/II COBOL
Operating Guide.

4. The L(evel) Command

Animation normally traces execution into any level of nested PERFORM statements. The Level command allows you to set a "threshold" at any level of nesting. After it is set, any PERFORM subordinate to this threshold level is treated as a single statement by ANIMATOR. The cursor is not moved into PERFORMed procedures below the threshold level.

Press L and the following subordinate menu is displayed:

PERFORM LEVEL = , THRESHOLD LEVEL = S(et) U(nset) E(xit) Q(uit)

CHAPTER 3. OPERATING COMMANDS

Note that the display indicates both the current PERFORM level and the threshold level currently set.

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.
- Q causes immediate abandonment of the current PERFORM without executing further statements. This allows tidy exit from a closed loop in a PERFORMed procedure.

5. 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, you must first place the cursor at the desired position.

Press P and the following subordinate menu is displayed:

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

Press the appropriate key:

- W repositions the screen window as necessary and positions the cursor at the next statement to be

CHAPTER 3. OPERATING COMMANDS

executed. This is useful as a check after you use the source screen manipulation commands, but note that it is not necessary since ANIMATOR will resume execution at the correct position unless you use the "R" command described below.

You may also enter this command on the main prompt line without the preceding P command.

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

6. The C(ompile) Command

This command enables immediate compilation and execution of a specified COBOL statement (or statements).

When you press C, the cursor is positioned for entry of the required COBOL statements followed by RETURN.

If you enter multiple statements, they should not be separated by periods. You may enter a final terminating period, but if omitted it is assumed. The entered syntax is not retained and thus will be executed once only.

If a syntax error is found in the entered statements, the following prompt is displayed:

```
COMPILER ERROR NO. nnn - S(top) C(ontinue)
```

CHAPTER 3. OPERATING COMMANDS

Press S to stop, or C to regain control.

Compiler error numbers are detailed in the L/II COBOL Operating Guide.

NOTE: ANIMATOR cannot handle certain complex statements (e.g. INSPECT). These will return a compiler error 1, and you will have to enter simpler statements to achieve the same effect.

7. The N(ame) Command

Within a hierarchy of programs, ANIMATOR will by default animate all programs compiled for animation. Other programs are executed normally.

This command allows finer control over which programs are animated.

Press N and the following subordinate menu is displayed:

PROGRAM NAME - W(hich) A(ny) T(his) O(ther)

Press the appropriate key, where:

W displays the current program name.

CHAPTER 3. OPERATING COMMANDS

A causes any program compiled for animation to be animated.

NOTE: this is the initial default.

T instructs ANIMATOR to animate only the program currently executing. Any others will be executed normally.

O Positions the cursor for entry of the name of the next program to be animated followed by RETURN. This must be entered as an alphanumeric literal, i.e. within quotes. Note: Animation of the current program continues until it completes, all others except the named program will be executed normally.

NOTE: If you wish to start the run by executing normally until a specified program is reached, use the following command-line directive when loading ANIMATOR:-

ZOOM (program name)

D. Examination and Amendment of Data

These commands provide facilities for the examination and amendment of the contents of specified data items. You can indicate selected data items by either specifying the dataname or pointing to one with the cursor.

1. The D(isplay) Command

This command allows you to display/amend a referenced data item by typing the name. To use this command press D, position the cursor for entry of the required data-name, then enter RETURN.

CHAPTER 3. OPERATING COMMANDS

The value of the specified data item is displayed with appropriate conversion in accordance with its PICTURE clause. Non-ASCII characters in alphanumeric data items are displayed as `^C`. Signs in numeric data items are displayed leading or trailing as appropriate, and decimal-points are inserted if used.

After the data item is displayed, the cursor is positioned to the next line to facilitate entry of a replacement value if required. If no replacement is required, press RETURN; otherwise, enter the new value in COBOL literal format (remember that 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 can be displayed at one time. For longer data items the remainder of the field can be displayed by specifying "data-name + n" where n is a numeric offset into the data item. Any amendments will only be applied to the part of the data item currently displayed.

2. The Q(uey) Command

This command allows you to display/amend a referenced data item by pointing to it with the cursor.

To use this command, position the cursor to any occurrence of the data-name within the source code, then press Q.

CHAPTER 3. OPERATING COMMANDS

The data item is displayed and may be amended as described for the D(isplay) command, with the exception that only the first 80 characters of long data items can be referenced.

3. The M(onitor) Command

This command enables automatic 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 you press M.
- U unsets the monitor.
- N positions the cursor for entry of the required data-name followed by RETURN. An offset may be specified - see the D(isplay) command.

The monitored item is redisplayed after each animation "step", for example, after each statement if in "GO" mode.

4. User Screen Display

During animation any ANSI format ACCEPT/DISPLAY statements are diverted to the bottom of the screen so that they do not

CHAPTER 3. OPERATING COMMANDS

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 you can interact with the screen in the normal way.

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

5. The U(ser) Command

Press U and the user screen is displayed, replacing the source code window display. This remains on the screen until any key is pressed, which reverts to the source code display.

APPENDIX A

ANIMATOR COMMAND SUMMARY

This section 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 Animation is invoked. A brief summary of each command is given.

A - The LocAte command enables the declaration of any data-name or procedure-name, specified by typing the name, to be located by moving the cursor to it.

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 five options:

S - Set breakpoint at statement pointed to by the cursor.

U - Unset the breakpoint currently pointed to by the cursor.

C - Cancel all breakpoints.

X - Examine next breakpoint. Can be used successively to examine all set breakpoints.

I - Set a breakpoint at statement currently pointed to by the cursor with an associated condition.

C - The Compile command enables immediate compilation and execution of an inserted COBOL statement (or statements) during Animation. If an error is made a S(top) or C(ontinue) option is offered.

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.

APPENDIX A. ANIMATOR COMMAND SUMMARY

- K - SKips a single COBOL statement without execution 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 the DELETE (or RUBOUT) key followed by a numeric character 1 through 9, where 1 is slowest and 9 is fastest.
- Z - Initiates executions 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.

APPENDIX A. ANIMATOR COMMAND SUMMARY

- U - Unsets the Monitor.
- N - Positions the cursor for operator entry of the Name of the data-item to be Monitored.
- N - The Name command enables detailed specification of which programs in a hierarchy are Animated.
- O - The locate command enables the declaration of any data-name or procedure-name, specified by moving the cursor to any occurrence thereof in the source listing, to be located by ANIMATOR moving the cursor to the declaration.
- 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. (> from main menu)
 - P - Displays Previous screen from source text. (< from main menu)
 - 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 the screen in Half (i.e. into two windows) with a dividing line of hyphens. The lower window is

APPENDIX A. ANIMATOR COMMAND SUMMARY

positioned to show the top of source text. Note: Subsequent screen commands operate in the window in which the cursor is positioned.

- F - Restores Full screen display (single window).
- =n - repositions the window such that the nth 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.
- T - The unTil command enables the user to specify a COBOL conditional expression which, if it becomes true, will cause execution to halt.

Upon entering T, three options are appended:

- S - Position the cursor for entry of the conditional expression.
- U - Cancel the previously set conditional expression.
- X - Displays any previously set conditional expression.
- U - The User command displays the current user screen replacing the source code window until any key is pressed.