

*Installation and
Operation
Manual*

**B 1000
Systems
COBOL74B
Test and Debug
System**

(Relative to Mark 12.0 System Software Release)
Copyright © 1985, Burroughs Corporation, Detroit, Michigan 48232

BURROUGHS CORP.
PRODUCT ASSURANCE SUPPORT
6300 HOLLISTER AVE.
GOLETA, CA. 93117

Burroughs cannot accept any financial or other responsibilities that may be the result of your use of this information or software material, including direct, indirect, special or consequential damages. There are no warranties extended or granted by this document or software material.

You should be very careful to ensure that the use of this software material and/or information complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Comments or suggestions regarding this document should be submitted on a Field Communication Form (FCF) with the CLASS specified as 2 (S.SW: System Software), the Type specified as 3 (DOC), and the product specified as the 7-digit form number of the manual (for example, 1177987).

LIST OF EFFECTIVE PAGES

Page	Issue
Title	Original
ii	Original
iii	Original
iv	Blank
v thru vii	Original
viii	Blank
1-1 thru 1-3	Original
1-4	Blank
2-1 thru 2-4	Original
3-1	Original
3-2	Blank
4-1 thru 4-38	Original
A-1 thru A-3	Original
A-4	Blank
1 thru 2	Original

TABLE OF CONTENTS

Section	Title	Page
1	INTRODUCTION	1-1
	General	1-1
	Manual Structure	1-1
	Related Documents	1-1
	Generic Terms	1-2
2	OVERVIEW	2-1
	COBOL74B/AID Screen Format	2-2
	Welcome Screen Format	2-2
	Main Operation Screen Format	2-2
	OUTPUT Field	2-3
	. FORMAT Field	2-3
	BREAKPOINT TABLE Area	2-3
	Review Screen Format	2-4
3	INSTALLATION	3-1
	Required Files	3-1
	Dynamic Memory Setting	3-1
	Processor Priority	3-1
	SMCS Jobs File Entry	3-1
4	OPERATION	4-1
	Executing the COBOL74B/AID Program	4-1
	Remote Files	4-1
	Terminating the COBOL74B/AID Program	4-1
	Operating the COBOL74B/AID Program	4-2
	COMMAND SET	4-2
	ABANDON	4-3
	ADOPT	4-4
	AT	4-5
	BACK	4-7
	BYE	4-8
	CLEAR	4-9
	CONTINUE	4-10
	DISCARD	4-12
	DISPLAY	4-14
	DO	4-16
	EXECUTE	4-17
	EXIT	4-18
	FORMAT	4-19
	FREQUENCY	4-20
	HELP	4-21
	HISTORY	4-22
	LOG	4-23
	MOVE	4-24
	NEXT	4-26
	OUTPUT	4-27
	PERFORM	4-28
	PRINT	4-29
	REVIEW	4-30
	SAME	4-31

TABLE OF CONTENTS (Cont)

Section	Title	Page
	STEP	432
	STOP	433
	STOP RUN	434
	TRAIL	435
	WHEN	436
	Plus (+), Minus (-)	438
A	SYNTAX AND NOTATION CONVENTIONS	A-1
	Required Items	A-1
	Optional Items	A-2
	Loops	A-2
	Bridges	A-3
	Keywords	A-3
	INDEX	1

LIST OF ILLUSTRATIONS

Figure	Title	Page
2-1	COBOL74B/AID Configuration	2-1
2-2	Welcome Screen Format	2-2
2-3	Main Operation Screen Format	2-3
2-4	Review Mode Screen	2-4
4-1	ABANDON Screen (MO)	4-1
4-2	ADOPT Screen (MO)	4-4
4-3	AT Screen (MO)	4-6
4-4	BACK Screen (R)	4-7
4-5	BYE Screen (MO)	4-8
4-6	CLEAR Screen (MO)	4-9
4-7	CONTINUE Screen (MO)	4-11
4-8	DISCARD Screen (MO)	4-13
4-9	DISPLAY Screen (MO)	4-15
4-10	DO Screen (MO)	4-16
4-11	EXECUTE Screen (MO)	4-17
4-12	EXIT Screen (MO)	4-18
4-13	FORMAT Screen (MO)	4-17
4-14	FREQUENCY Screen (MO)	4-20
4-15	HELP Screen (MO)	4-21
4-16	HISTORY Screen (MO)	4-22
4-17	LOG Screen (MO)	4-23
4-18	MOVE Screen (MO)	4-25
4-19	NEXT Screen (MO)	4-26
4-20	OUTPUT Screen (MO)	4-27
4-21	PERFORM Screen (MO)	4-28
4-22	PRINT Screen (MO)	4-29
4-23	REVIEW Screen (R)	4-30
4-24	SAME Screen (R)	4-31
4-25	STEP Screen (MO)	4-32
4-26	STOP Screen (MO)	4-33
4-27	STOP RUN Screen (MO)	4-34
4-28	TRAIL Screen (MO)	4-35
4-29	WHEN Screen (MO)	4-37
4-30	MINUS (-) Screen (R)	4-38

SECTION 1 INTRODUCTION

GENERAL

This manual describes the installation and operation of the COBOL74B Analyzer and Interactive Debugger program. Throughout this manual, this program is referred to as the COBOL74B/AID program.

The COBOL74B/AID program supplies a COBOL74B programmer with an interactive debugging tool. Using the COBOL74B/AID command set, the programmer can control the execution of a COBOL74B program. This includes displaying data, changing data fields, and manipulating the flow of the program. A history of the displayed data and breakpoints can be recorded for later viewing.

This manual is intended for COBOL74B programmers. It is assumed that the reader has a working knowledge of the basic hardware and software functions of the B 1000 processing systems.

MANUAL STRUCTURE

A brief description of each section and appendix of this manual follows.

Section 1: Introduction

Describes the purpose and structure of the manual, and provides a list of related documentation.

Section 2: Overview

Describes the features and capabilities of the COBOL74B/AID program for COBOL74B programmers.

Section 3: Installation

Describes how to install the COBOL74B/AID program.

Section 4: Operation

Describes the COBOL74B/AID program operating instructions and the COBOL74B/AID commands.

Appendix A: Syntax and Notation Conventions

Describes the syntax and notation conventions used in this manual.

RELATED DOCUMENTS

The following manuals, available from the Burroughs Corporation, are related to this manual.

B 1000 Systems COBOL74 Language Manual, form number 1168622.

B 1000 Systems System Software Operation Guide, Volume 1, form number 1169000.

B 1000 Systems SMCS Installation, Operation, and Functional Description Manual (12.0) form number 5016488.

GENERIC TERMS

A number of generic terms are used within this manual to describe the syntax of commands. These terms are defined in the following paragraphs.

<dataname>

A <dataname> is any data item in a COBOL74B program.

<constant>

A <constant> is any integer valid in a COBOL74B program.

not a <figurative constant>!

<condition-name>

A <condition-name> is any condition name in a COBOL74B program.

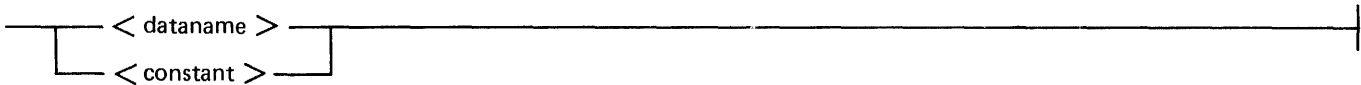
<procedure-name>

A <procedure-name> is any paragraph or section name in the COBOL74B program.

<value>

A <value> is any dataname or constant in a COBOL74B program.

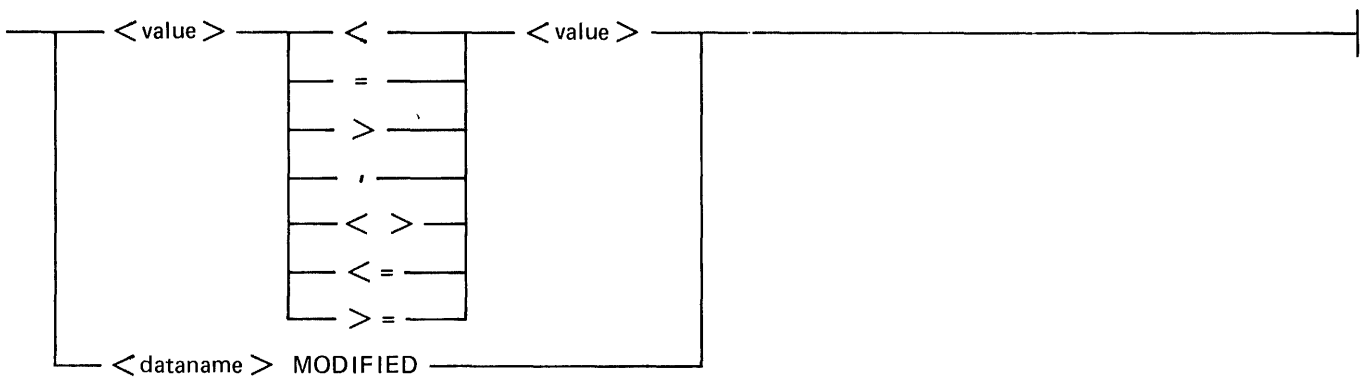
<value> syntax:



<boolean-primary>

A <boolean-primary> allows two <dataname>s or a <dataname> and a constant to be compared. Using the keyword MODIFIED causes the COBOL74B/AID program to check if <dataname> is modified.

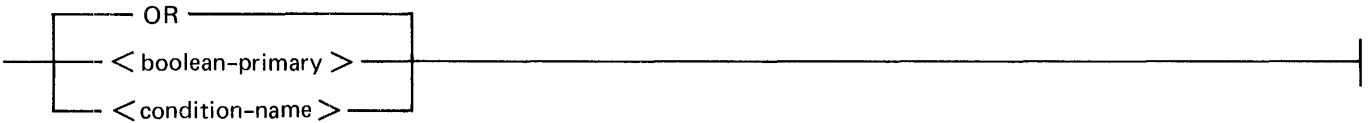
<boolean-primary> syntax:



<boolean-expression>

A <boolean-expression> allows a <boolean-primary> or a <condition-name> to be part of an expression.

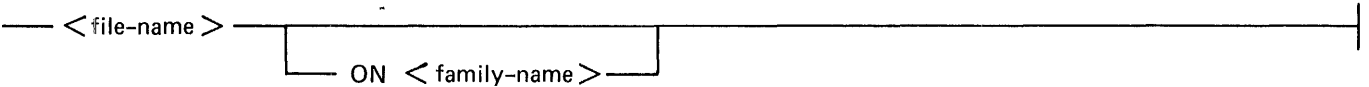
<boolean-expression> syntax:



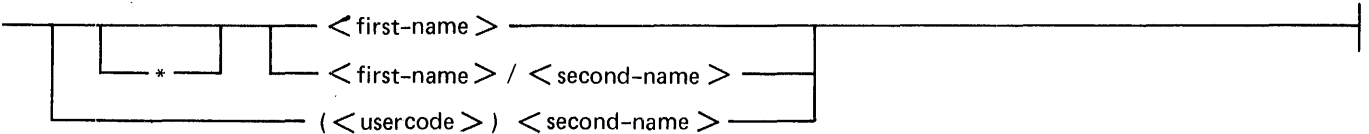
<title>

A <title> references a file and the name of the disk on which it resides.

<title> syntax:



<file-name> syntax:



<title> semantics:

<family-name>

A <family-name> is the name of a disk or pack and consists of one to ten characters.

*

The asterisk character overrides the default usercode and the default pack name associated with that usercode.

<first-name>

The <first-name> is the first part of a <file-name> and consists of one to ten characters.

<second-name>

A <second-name> is the second part of a <file-name> and consists of one to ten characters.

<usercode>

<usercode> is a name assigned to a user to secure system and file access.

SECTION 2 OVERVIEW

The COBOL74B/AID program supplies a COBOL74B programmer with an interactive debugging tool. When the COBOL74B/AID program is executed, it opens a remote file to communicate with the programmer. At this time the programmer can execute a COBOL74B program, display data, change data, and control the program's execution with COBOL74B/AID commands. The following symbolic diagram shows the relationship of the COBOL74B/AID program, the COBOL74B program, and the other system software. Note how the COBOL74B/AID program actually communicates with the special COBOL74B/DEBUG interpreter and refers to the COBOL74B program disk file for the symbol table information.

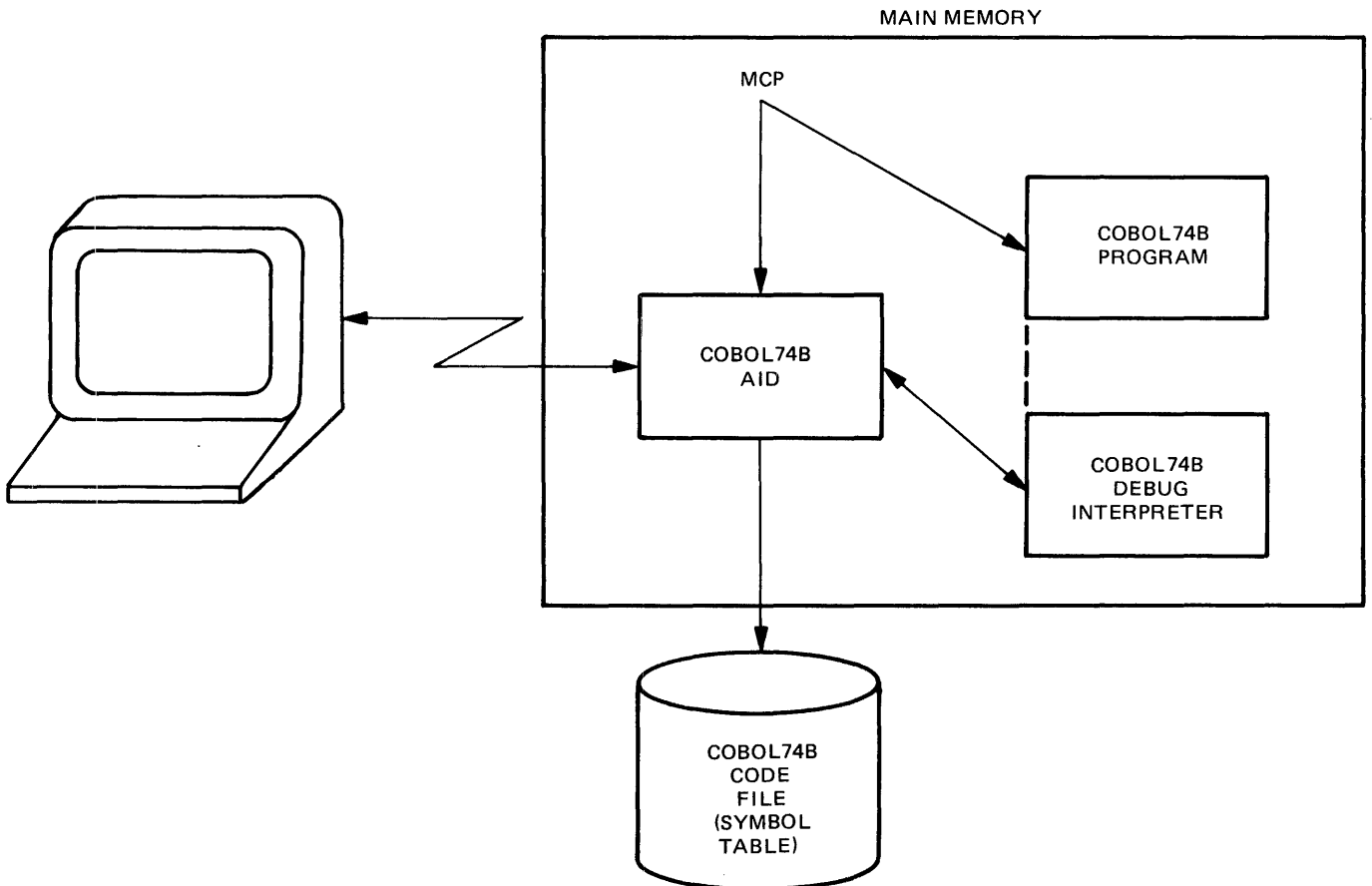


Figure 2-1. COBOL74B/AID Configuration

COBOL74B/AID SCREEN FORMAT

The COBOL74B/AID program has three different screen formats to provide communication with the COBOL74B programmer. The screen formats are welcome, main operation, and review. Each format has a specific purpose, as implied by the names, and each has a status line located at the bottom of the screen. The following paragraphs describe the three formats.

Welcome Screen Format

After the COBOL74B/AID program is executed, a welcome screen is displayed on the terminal. In figure 2-2, note the status line at the bottom of the screen that reads NO JOB RUNNING. The COBOL74B program to be analyzed does not execute until specified with an EXECUTE or an ADOPT command (described in section 4). After a new screen is displayed, the welcome screen is not displayed again.

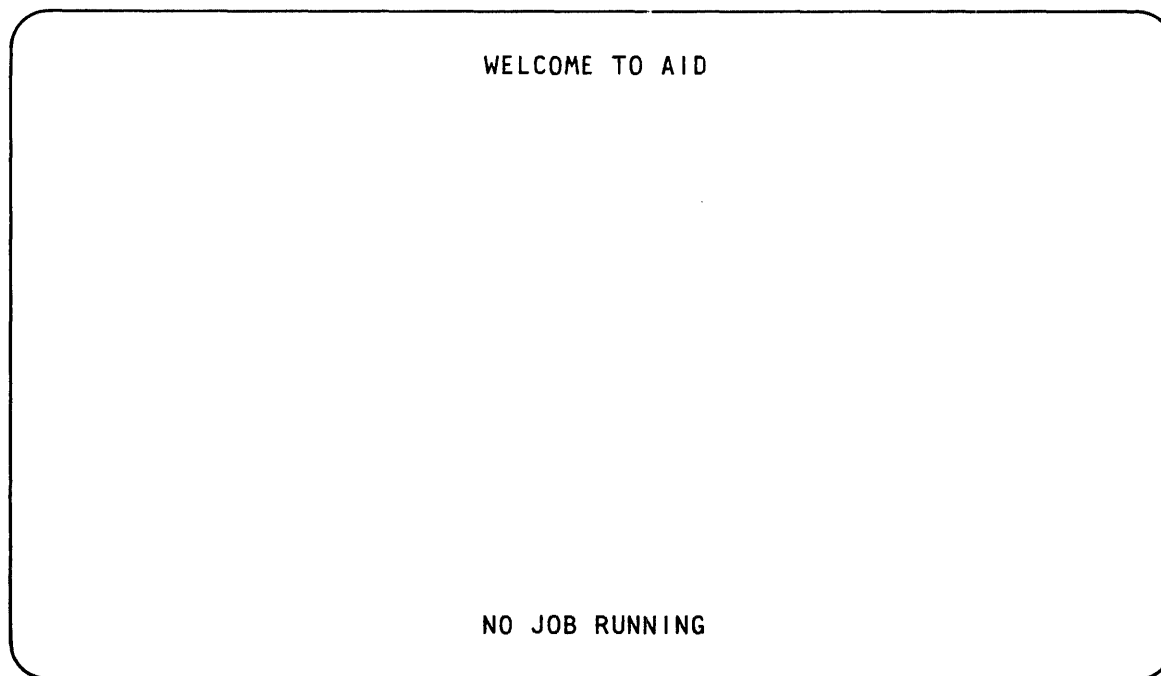


Figure 2-2. Welcome Screen Format

Main Operation Screen Format

The main operation screen, the most common format, is used to display data items and a portion of the trace buffer. Also, most of the COBOL74B/AID commands are entered here.

From the welcome screen, the main operation screen appears when the EXECUTE command is entered or the SPCFY key is pressed. From any other screen, the main operation screen appears when the CLEAR command is entered or the SPCFY key is pressed.

The following paragraphs describe the components of a main operation screen. Figure 2-3 shows the format.

OUTPUT Field

The OUTPUT field specifies the output medium for the data generated by the COBOL74B/AID program. Refer to the OUTPUT command in section 4 for a complete description of the OUTPUT field.

FORMAT Field

The FORMAT field specifies the type of format used to display data. The format is either edited for display purposes or the hexadecimal representation. Refer to the FORMAT command in section 4 for a complete description of the FORMAT field.

BREAKPOINT TABLE Area

The BREAKPOINT TABLE lists the breakpoints specified by the COBOL74B programmer. Breakpoints are paragraph names, code addresses, or conditions that the COBOL74B programmer wants the COBOL74B/AID program to recognize as breakpoints. Each time a breakpoint location is reached in the COBOL74B program, the COBOL74B program is suspended and can be analyzed with the COBOL74B/AID program. For example, in figure 2-3, a program called ACCOUNT/PAYROLL is being analyzed, and three breakpoints are established. In this case, these breakpoints are paragraph names in the ACCOUNT/PAYROLL program.

Also in figure 2-3, a temporary breakpoint, STOP, is identified by a hyphen (-) character. Temporary breakpoints are removed from the breakpoint table after the CONTINUE command is entered. The highlighted breakpoint, STOP, is the current breakpoint.

```
OUTPUT: REMOTE                FORMAT: EXTERNAL
                                BREAKPOINT TABLE
1                               : AT CREATE-FIGURES STOP
2 HEADING                       : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP
3                               : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...
█                               : STOP

ACCOUNT/PAYROLL ON PACKZ (#565) AT PRINT-DETAIL
```

Figure 2-3. Main Operation Screen Format

Review Screen Format

Executing the REVIEW command causes review mode to be entered and the review screen format to be displayed. In this format, the last 22 entries of the trace buffer can be displayed. The trace buffer contains the data displayed below the breakpoint table on the main operation screen. The trace buffer is limited to 256 lines of data and is maintained in a wrap-around fashion. The PAGE, NEXT, BACK, and SAME commands are valid in review mode. These commands control the buffer output on the screen. The CLEAR command is also valid during review mode and causes the main operation screen to be refreshed. Refer to section 4 for a complete description of the REVIEW, PAGE, NEXT, BACK, SAME, and CLEAR commands. Figure 2-4 shows the review mode screen format.

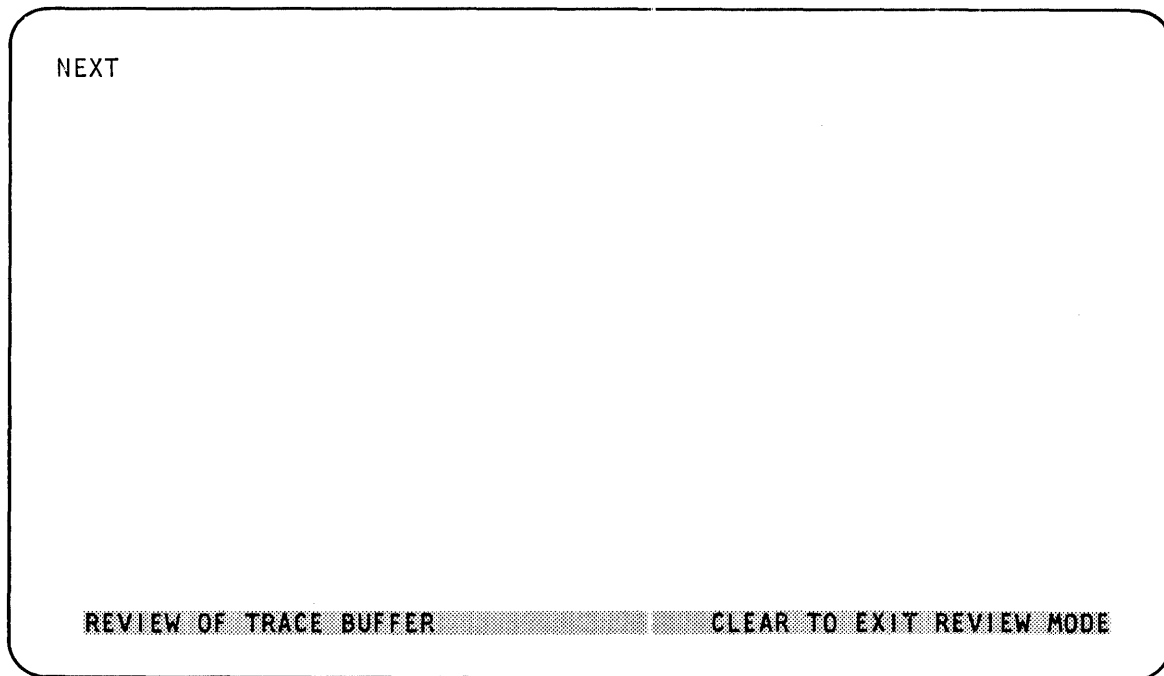


Figure 2-4. Review Mode Screen

SECTION 3 INSTALLATION

This section describes the installation procedures for the COBOL74B/AID program. Before the COBOL74B/AID program is executed, the required files, dynamic memory size, processor priority, and entry into the SMCS JOBS file must be considered.

REQUIRED FILES

The COBOL74B/AID, COBOL74B/AID-GRAMR, COBOL74B/AID-TEACH, COBOL74B/DEBUG1M, and the COBOL74B/INTERP files must be copied to system disk before the COBOL74B/AID program can be executed.

DYNAMIC MEMORY SETTING.

The dynamic memory requirement for the COBOL74B/AID program is set to 25000 bytes at release time. This is the recommended setting.

PROCESSOR PRIORITY

For optimum response, the COBOL74B/AID program must run at a higher priority than other programs in the mix, but at a lower priority than the text editors, such as the CANDE program. For instance, if a system compiles and executes programs at a priority of 4 and executes the text editor at priority 13, then the recommended priority for the COBOL74B/AID program is 12. The following example shows how to permanently assign a priority of 12 to the COBOL74B/AID program.

Example:

```
MODIFY COBOL74B/AID PRIORITY 12;
```

SMCS JOBS FILE ENTRY

The COBOL74B/AID program can be entered in the SMCS/JOBS file. The following is an example of an entry for the COBOL74B/AID program. For systems that use the Work Flow Language (WFL) compiler option, the EXECUTE instruction must be replaced with the RUN command.

Example:

```
$COBOL74B/AID  
NO EOF LOG ON EXCEPT NO-SESSION;  
EXECUTE COBOL74B/AID;PRIORITY 12;
```

For a detailed explanation of the SMCS JOBS file and its creation, refer to the *B 1000 Systems SMCS Installation, Operation, and Functional Description Manual*.

SECTION 4 OPERATION

This section describes the operation procedures for the COBOL74B/AID program.

EXECUTING THE COBOL74B/AID PROGRAM

The COBOL74B/AID program can be initiated from a remote station. Program execution is performed by entering one of the following command strings from a remote terminal.

```
EXECUTE COBOL74B/AID
```

```
EX COBOL74B/AID
```

Remote Files

The COBOL74B/AID program opens a remote file to communicate with a terminal. If the COBOL74B program to be analyzed also opens a remote file, the COBOL74B program should either be executed from within the COBOL74B/AID program (as opposed to adopting it), or executed from a different terminal. The COBOL74B/AID program is designed to allow another program to open its remote file to the same terminal, but the COBOL74B programs may not be designed to allow the COBOL74B/AID program to open its remote file.

When a COBOL74B/AID program that opens a remote file is executed from the COBOL74B/AID program, the COBOL74B/AID program displays its data on page one and the COBOL74B program's data is displayed on page two. After the COBOL74B/AID program and the COBOL74B program open remote files, a colon (:) signal character must precede all COBOL74B/AID program commands. This is to distinguish between the data for the COBOL74B program and the COBOL74B/AID program commands.

To execute a COBOL74B program that opens a remote file outside of the COBOL74B/AID program control, the programs must be executed from two different terminals. This is also the requirement when a COBOL74B program with an open remote file is waiting on a DS/DP/ADOPT condition.

TERMINATING THE COBOL74B/AID PROGRAM

The COBOL74B/AID program may be terminated by entering the BYE or END command if no program is running or by entering the STOP RUN command followed by the BYE command if a program is running.

OPERATING THE COBOL74B/AID PROGRAM

There are four basic steps involved in a session with the COBOL74B/AID program.

1. First, the EXECUTE or ADOPT command can be entered to establish the COBOL74B program to be analyzed. A beginning of job message is displayed on the screen, and the COBOL74B program is suspended at the first paragraph name. This is a temporary breakpoint.
2. Next, paragraph names, code addresses, and conditions can be specified as breakpoints with the AT and WHEN commands. Each time the COBOL74B program control reaches a breakpoint, the COBOL74B/AID program highlights the breakpoint name in the table and suspends the COBOL74B program.
3. After breakpoints are established, the programmer can enter CONTINUE and the COBOL74B program continues processing until another breakpoint or end of job occurs. Also, during a breakpoint, several commands, as described in this section, can be entered.
4. Finally, to suspend the COBOL74B program, the STOP command can be entered. Also, the program can be forced to go to end of job with the STOP RUN command. If the status is NO JOB RUNNING, the COBOL74B/AID program can be terminated with the BYE or END command.

COMMAND SET

The following pages describe the COBOL74B/AID command set. Note the status that applies for each command. For example, some commands are only valid when the status is NO JOB RUNNING, while others are only valid when the status is JOB IS RUNNING.

The first 80 characters of a command string are valid. Any part of the string that exceeds 80 characters is truncated and causes an error message to be displayed.

Only five commands, CLEAR, PAGE, NEXT, BACK, and SAME, are valid in review mode. All other commands are invalid during review mode. If any are used, an error message is displayed.

The example program used in this section is called ACCOUNT/PAYROLL. The first paragraph name in the ACCOUNT/PAYROLL program is OPEN-FILES. Therefore, many of the examples have a status message of ACCOUNT/PAYROLL ON PACKZ (#565) AT OPEN-FILES. Since OPEN-FILES is the first paragraph encountered, it is also the first temporary breakpoint.

NOTE

In the captions to the figures 4-1 through 4-30, (MO) means main operation screen format and (R) means review screen format.

ABANDON

The ABANDON command releases the current COBOL74B program from COBOL74B/AID program control. After this command is accepted, the status becomes NO JOB RUNNING. This command is only valid during a breakpoint or when the COBOL74B program is executing.

Syntax:

— ABANDON —

Figure 4-1 is the screen image after the ABANDON command is accepted.

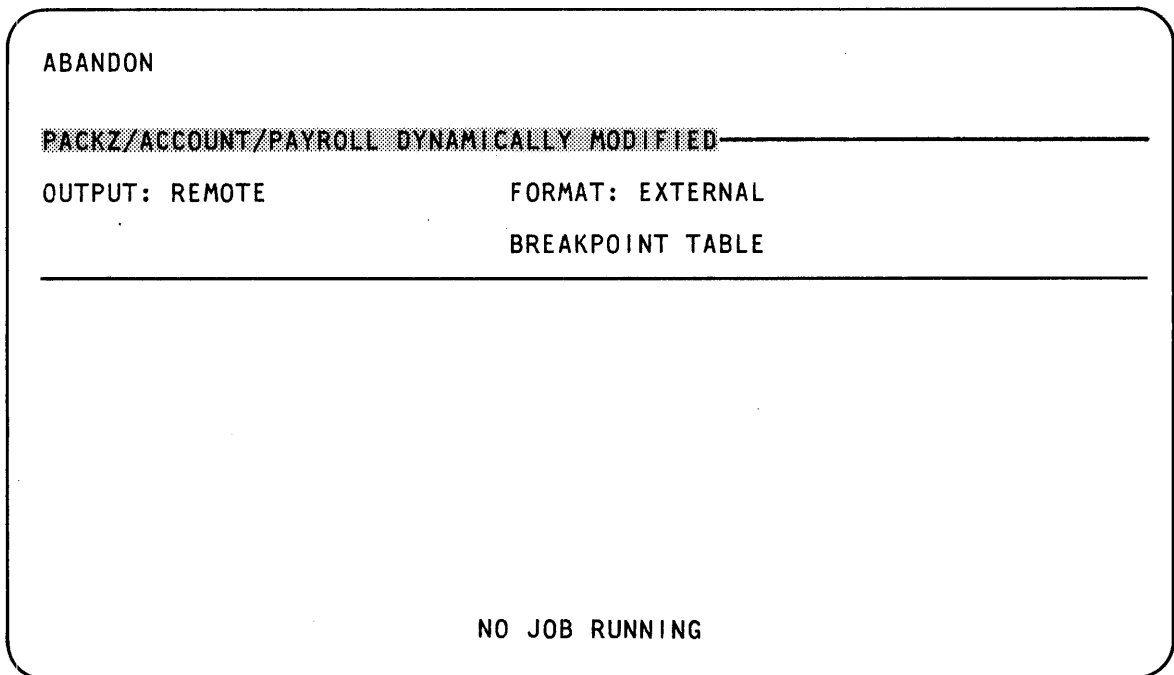


Figure 4-1. ABANDON Screen (MO)

ADOPT

The ADOPT command establishes control with a COBOL74B program that is in the job mix. Once this command is accepted, the status becomes JOB RUNNING. The ADOPT command is only valid when the status is NO JOB RUNNING.

For security purposes, there are restrictions on the COBOL74B programs that can be adopted. The \$ TADS COBOL74B compiler option prevents COBOL74B programs from becoming adopted if the option is reset. If, however, the option is set or not specified, then the program can be adopted. Also, the NO.ADOPT program control attribute can be applied to a COBOL74B object file. This attribute prevents a program from becoming adopted, it cannot be used to make the program adoptable; NO NO.ADOPT is not a valid option.

Note that COBOL74B programs that are allowed COBOL74B/AID program analysis wait on a DS/DP/ADOPT state instead of DS/DP only. At this time, as well as any other time, the unprotected COBOL74B program can be adopted and analyzed.

Syntax:

— ADOPT <title> <job-number> —————|

Semantics:

<job-number>

Specifies the job number of the COBOL74B program in the job mix to be adopted.

Figure 4-2 shows an ADOPT screen.

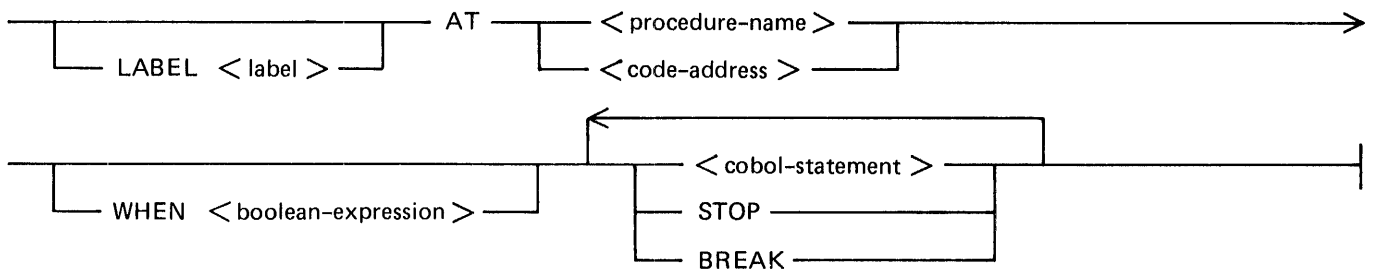
```
ADOPT 565  
  
LOADING SYMBOL TABLE INFORMATION  
-----  
OUTPUT: REMOTE           FORMAT: EXTERNAL  
                          BREAKPOINT TABLE  
-----  
  
NO JOB RUNNING
```

Figure 4-2. ADOPT Screen (MO)

AT

The AT command enters a breakpoint into the breakpoint table. A breakpoint is a paragraph name, code address, or condition in the COBOL74B program. Each time the COBOL74B program control reaches the breakpoint, the <cobol-statement> or STOP/BREAK commands are performed. The WHEN command provides a conditional clause within the AT command. The AT command is only valid during a breakpoint or when the COBOL74B program is suspended.

Syntax:



Semantics:

LABEL

The LABEL keyword associates a <label> field with the breakpoint.

<label>

The <label> field specifies a label for a breakpoint. This field must be an identifier valid in the COBOL74B language and can be repeated for any number of breakpoints. The <label> field is truncated after the tenth character.

<procedure-name>

The <procedure-name> field must be a valid paragraph or section name in the COBOL74B/AID program.

<code-address>

The <code-address> field is obtained from the COBOL74B program listing. The right side of the listing contains the segment and displacement numbers, separated by a comma.

<code-address> syntax:

```

    # <segment-number>, <displacement-number>
    
```

WHEN

The WHEN keyword specifies that a <boolean-expression> is to be evaluated. The <boolean-expression> must be true for the COBOL74B program to STOP or for the <cobol-statement> to be performed.

<boolean-expression>

The <boolean-expression> field specifies the conditions for a breakpoint to occur. Refer to Generic Terms in section 1 for a complete description of the <boolean-expression> field.

B 1000 Systems COBOL74 Test and Debug System
Installation and Operation Manual
OPERATION

<cobol-statement>

The <cobol-statement> field allows the use of the DISPLAY, MOVE, and STOP RUN commands within the AT command. Refer to the individual descriptions of these commands in this section for more information.

STOP

Using the STOP keyword suspends the COBOL74B program at the specified breakpoint. Control is given to the COBOL74B/AID program.

BREAK

The BREAK keyword is synonymous with the STOP keyword described previously.

Examples:

AT CREATE-FIGURES STOP

LABEL HEADING AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCVS-H-1 STOP

AT PRINT-DETAILS WHEN TEST-ID = TEST-CONTROL MOVE TEST-PERIOD TO TEST-ID

AT #0, 9416 STOP

Figure 4-3 displays the results of entering the above commands.

```
OUTPUT: REMOTE                FORMAT: EXTERNAL
                                BREAKPOINT TABLE
1                               : AT CREATE-FIGURES STOP
2 HEADING                      : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP
3                               : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...
4                               : AT #0, 9416 STOP

ACCOUNT/PAYROLL ON PACKZ (#565) AT OPEN-FILES
```

Figure 4-3. AT Screen (MO)

BACK

The BACK command displays the previous page of the trace buffer. This command is valid only in REVIEW mode. Refer to the REVIEW command in this section for a complete description of the REVIEW mode.

Syntax:

— BACK —

Figure 4-4 displays the results of entering BACK.

```
NEXT
AT PRINT-DETAIL REC-CT = 0
AT PRINT-DETAIL REC-CT = 5
AT COMPUTE-TOTALS WORK-HOURS = 40

REVIEW OF TRACE BUFFER CLEAR TO EXIT REVIEW MODE
ACCOUNT/PAYROLL ON PACKZ (#565) AT COMPUTE-TOTALS
```

Figure 4-4. BACK Screen (R)

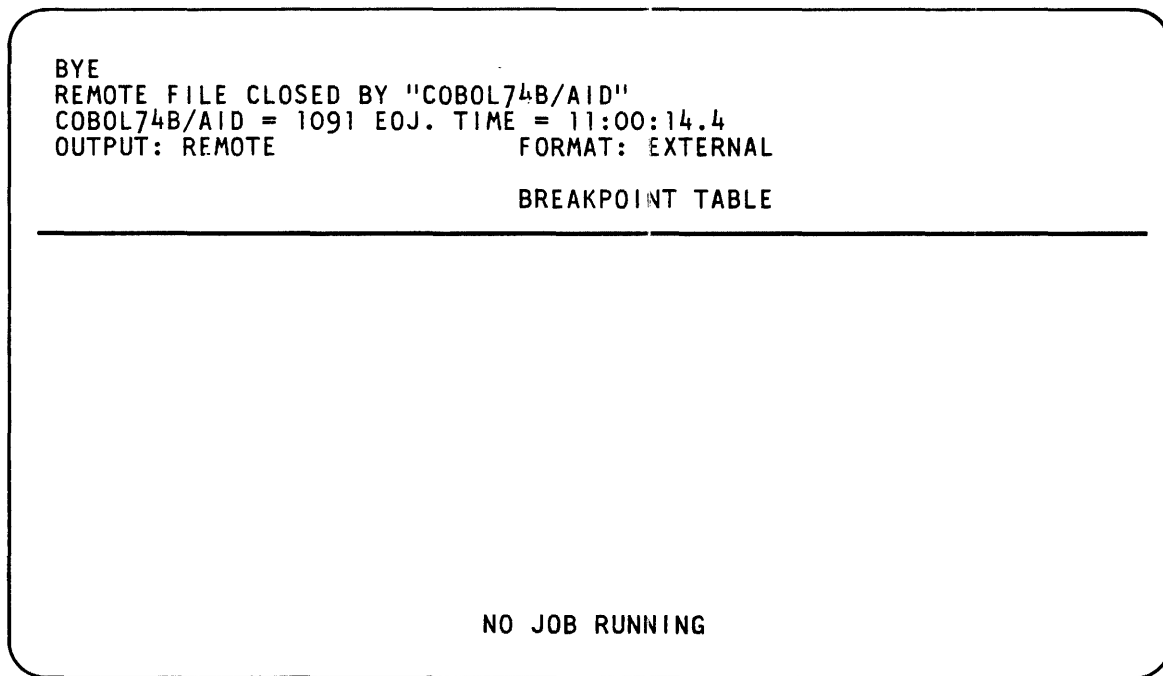
BYE

The BYE command terminates the COBOL74B/AID program. This command is only valid when the status is NO JOB RUNNING.

Syntax:

— BYE —

Figure 4-5 shows the screen.



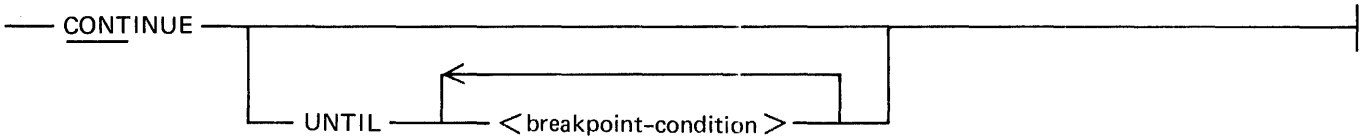
```
BYE
REMOTE FILE CLOSED BY "COBOL74B/AID"
COBOL74B/AID = 1091 EOJ. TIME = 11:00:14.4
OUTPUT: REMOTE          FORMAT: EXTERNAL
                          BREAKPOINT TABLE
-----
NO JOB RUNNING
```

Figure 4-5. BYE Screen (MO)

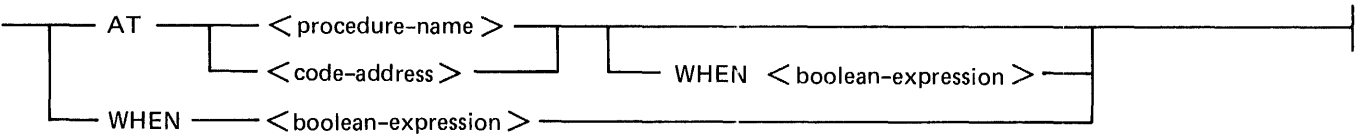
CONTINUE

The CONTINUE command causes the COBOL74B program to continue processing until the next breakpoint or the end of job occurs. Temporary breakpoints can be specified using the UNTIL keyword. The CONTINUE command is only valid during a breakpoint.

Syntax:



< breakpoint-condition > syntax:



Semantics:

UNTIL

The UNTIL keyword enters a temporary breakpoint into the breakpoint table. If the < breakpoint-condition > occurs, the break is performed, and the entry is removed from the breakpoint table.

< breakpoint-condition >

The < breakpoint-condition > field specifies the condition necessary for a break to occur. Refer to Generic Terms in section 1 for a complete description of the semantics used in the < breakpoint-condition > syntax.

Figure 4-7 shows a typical screen.

```
CONTINUE UNTIL AT MPY-TOTALS

-----
OUTPUT: REMOTE                FORMAT: EXTERNAL
                               BREAKPOINT TABLE
1                               : AT CREATE-FIGURES STOP
2 HEADING                      : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP
3                               : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...
4                               : AT #0, 9416 STOP
☒ CONTINUE UNTIL AT MPY-TOTALS

-----

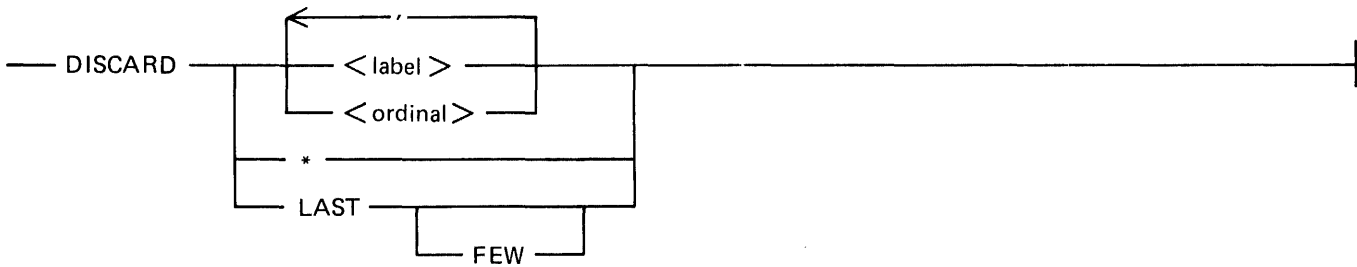
ACCOUNT/PAYROLL ON PACKZ (#565) AT MPY-TOTALS
```

Figure 4-7. CONTINUE Screen (MO)

DISCARD

The DISCARD command removes an entry from the breakpoint table. Breakpoints can be removed individually or in groups as specified with keywords. The DISCARD command is only valid during a breakpoint.

Syntax:



Semantics:

<label>

Using the <label> field causes the breakpoint entry or entries associated with that label to be removed from the breakpoint table.

<ordinal>

Using the <ordinal> field causes the breakpoint entry associated with <ordinal> to be removed.

*

Using the asterisk (*) character causes all the entries in the breakpoint table to be removed.

LAST

Using the LAST keyword causes the last breakpoint entry to be removed.

LAST FEW

Using the LAST FEW keywords causes all the breakpoints entered since the current breakpoint to be removed.

Figure 4-8 shows a DISCARD screen.

```
DISCARD 1

-----

OUTPUT: REMOTE          FORMAT: EXTERNAL
                        BREAKPOINT TABLE
2 HEADING      : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP
3              : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...
4              : AT#0, 9416 STOP

-----

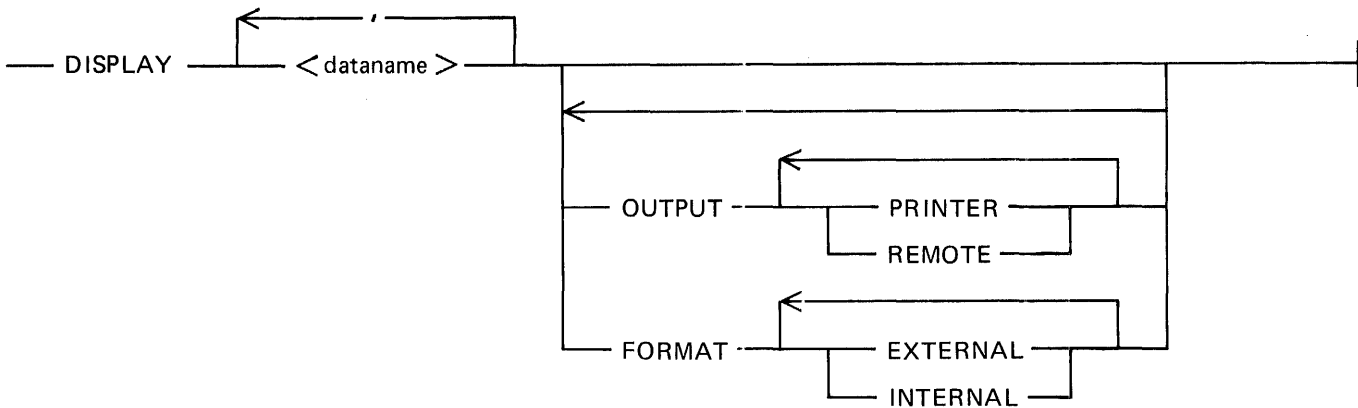
ACCOUNT/PAYROLL ON PACKZ (#565) AT PRINT-DETAIL
```

Figure 4-8. DISCARD Screen (MO)

DISPLAY

The DISPLAY command displays data items from the current COBOL74B program. The display medium can be a printer or remote device. The DISPLAY command is also valid in the context of the AT and WHEN commands described in this section. The DISPLAY command is only valid during a breakpoint.

Syntax:



Semantics:

< dataname >

The < dataname > field identifies the data item in the COBOL74B program to be displayed.

OUTPUT

The OUTPUT keyword specifies where the data item is to be displayed. If the OUTPUT keyword is omitted, the OUTPUT specification on the screen is used.

REMOTE

The REMOTE keyword specifies that the data item is to be displayed on the terminal device.

PRINTER

The PRINTER keyword specifies that the data item is to be printed on the printer device.

FORMAT

The **FORMAT** keyword specifies the type of representation used for the display of <dataname>. If the **FORMAT** keyword is omitted, the **FORMAT** specification on the screen is used.

EXTERNAL

Using the **EXTERNAL** keyword causes the data to be edited for display purposes.

INTERNAL

Using the **INTERNAL** keyword causes the data to be displayed in hexadecimal form.

Figure 4-9 shows a **DISPLAY** screen.

```
DISPLAY RECORD-COUNT

-----

OUTPUT: REMOTE                FORMAT: EXTERNAL
                                BREAKPOINT TABLE

 1                               : AT CREATE-FIGURES STOP
 2 HEADING                       : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP
 3                               : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...
 4                               : AT #0, 9416 STOP
 * DISPLAY RECORD-COUNT

-----

AT PRINT-DETAIL RECORD-COUNT=51
ACCOUNT/PAYROLL ON PACKZ (#565) AT PRINT-DETAIL
```

Figure 4-9. DISPLAY Screen (MO)

DO

The DO command causes a list of commands to be executed. This list is stored in a disk file that is created with the LOG command, described later in this section. The DO command is a macro facility that allows a session to be recorded and then recalled for future use. The commands are executed sequentially until the end of file is reached, an error condition occurs, or a command is entered at the terminal. Commands in the file are not processed when the COBOL74B program is executing. This file can be edited with a text editor such as the CANDE program. The DO command is valid during all modes of operation.

Syntax:

— DO — <title>

Semantics:

<title>

The <title> field identifies the disk file. Refer to Generic Terms in section 1 for a complete description of <title>.

Figure 4-10 shows a DO screen.

```
DO ACCOUNT/LOGFILE
-----
OUTPUT: REMOTE          FORMAT: EXTERNAL          PROCESSING DO FILE
                        BREAKPOINT TABLE
-----
NO JOB RUNNING
```

Figure 4-10. DO Screen (MO)

EXIT

The EXIT command forces an exit from the current paragraph in the COBOL74B program. A temporary entry in the breakpoint table is made. The EXIT command affects the program control only if a PERFORM statement is in effect. This command is only valid during a breakpoint.

Syntax:

— EXIT —

Figure 4-12 shows an EXIT screen

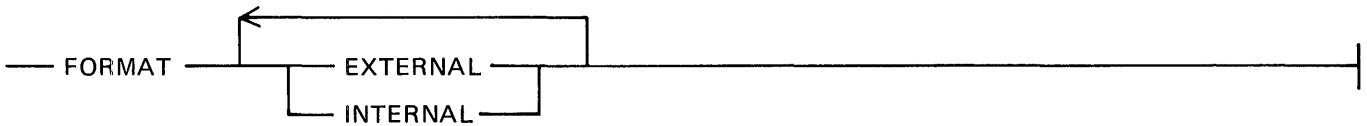
```
EXIT
-----
OUTPUT: REMOTE          FORMAT: EXTERNAL
                        BREAKPOINT TABLE
 1      : AT CREATE-FIGURES STOP
 2 HEADING : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP
 3      : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...
 4      : AT #0, 9416 STOP
  : EXIT
-----
ACCOUNT/PAYROLL ON PACKZ (#565) AT CREATE-FIGURES
```

Figure 4-12. EXIT Screen (MO)

FORMAT

The **FORMAT** command changes the display format to either **INTERNAL** or **EXTERNAL**. This command is also available in the context of the **DISPLAY** command described in this section. When the **FORMAT** field is explicitly stated in the **DISPLAY** command, the **FORMAT** specification on the screen is ignored. The **FORMAT** field defaults to **EXTERNAL**. The **FORMAT** command is only valid during a breakpoint.

Syntax:



Semantics:

EXTERNAL

Using the **EXTERNAL** keyword causes the data to be displayed in edited form.

INTERNAL

Using the **INTERNAL** keyword causes the data to be displayed in hexadecimal form.

Figure 4-13 shows a **FORMAT** screen.

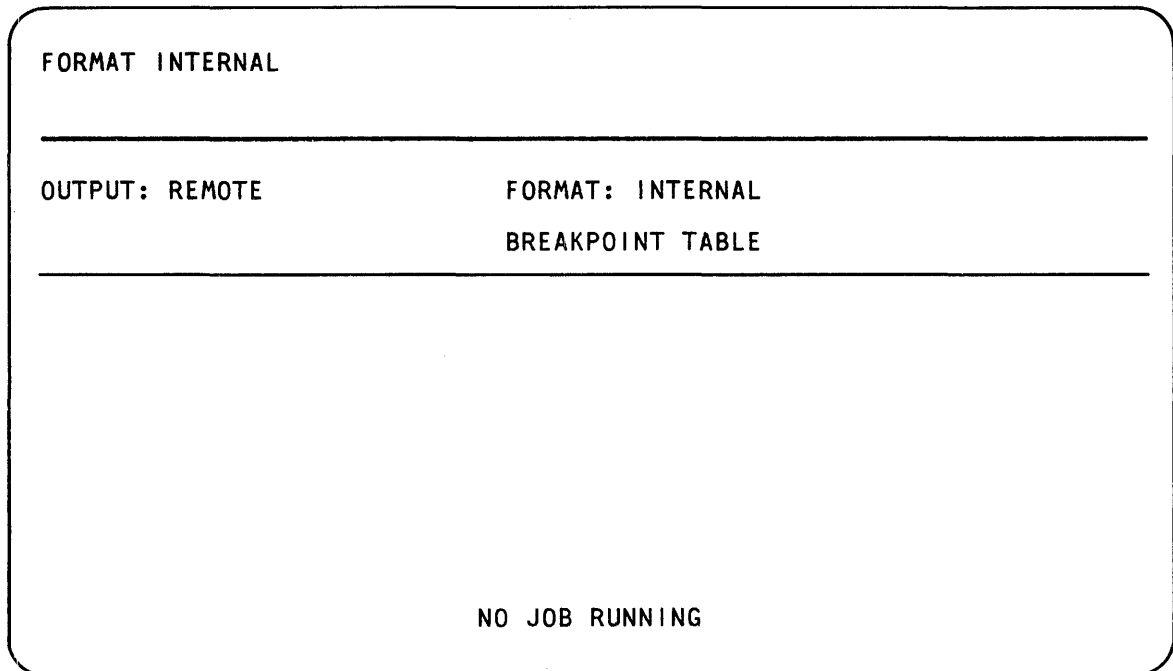
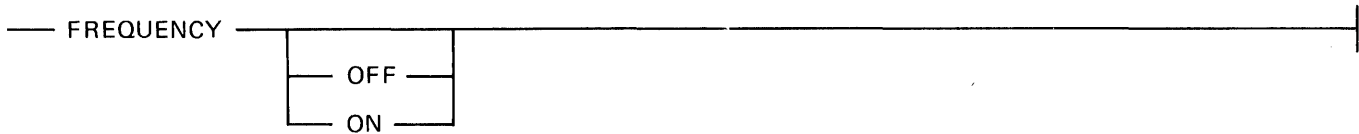


Figure 4-13. **FORMAT** Screen (MO)

FREQUENCY

The FREQUENCY command causes the COBOL74B/AID program to record the number of times each paragraph in the current COBOL74B program is entered. Each time the FREQUENCY command is entered, the FREQUENCY count is cleared, and the recording of paragraph entries begins again. The FREQUENCY data can be printed using the PRINT command described in this section. The FREQUENCY command is only valid during a breakpoint.

Syntax:



Semantics:

OFF

The OFF keyword resets the FREQUENCY boolean.

ON

The ON keyword sets the FREQUENCY boolean.

Figure 4-14 shows a FREQUENCY screen.

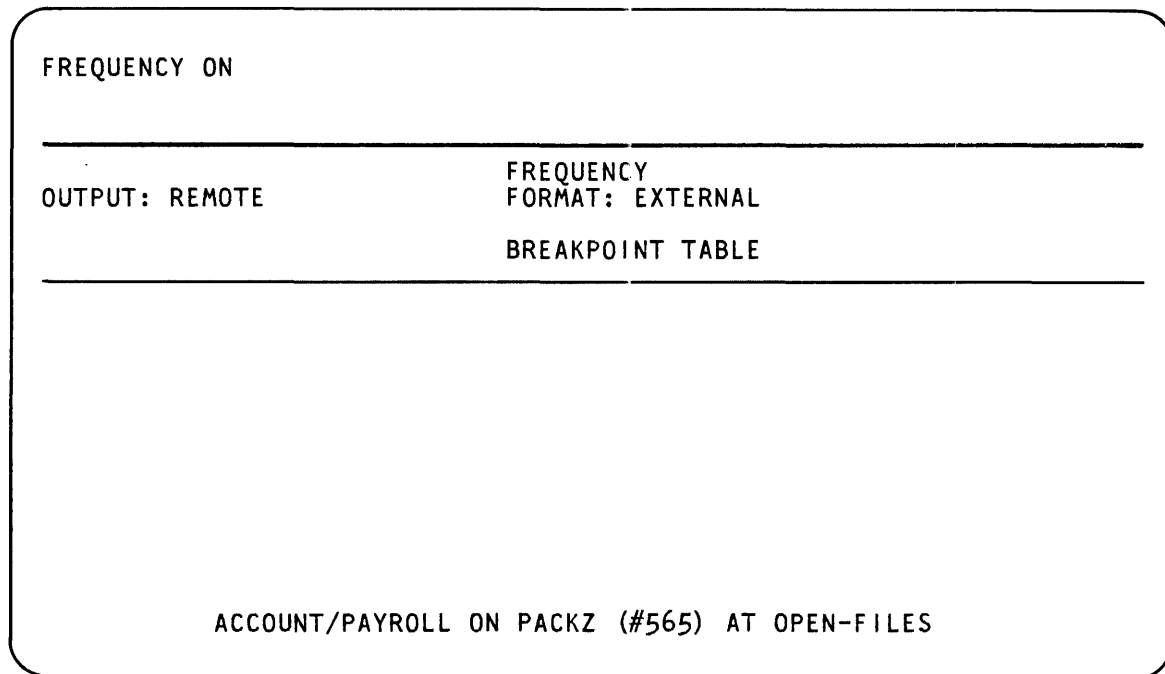
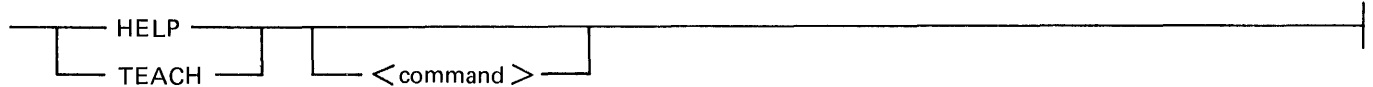


Figure 4-14. FREQUENCY Screen (MO)

HELP

The HELP command displays the COBOL74B/AID program commands. Specific, full-page command descriptions can be displayed by entering HELP followed by the command. This command is valid during all modes of operation.

Syntax:



Semantics:

TEACH

The TEACH keyword is synonymous with the HELP keyword.

<command>

The <command> field specifies any COBOL74B/AID command. A full-page description of this command is displayed when <command> is specified. The command description remains on the screen until a valid command is entered.

Figure 4-15 shows a HELP screen.

```

HELP
                                B1000 COBOL74B/AID COMMANDS:
ABANDON      CLEAR      EXECUTE      LOG      PRINT      TEACH
ADOPT        CONTINUE   EXIT        MOVE     REVIEW     TRAIL
AT           DISPLAY    FORMAT     NEXT     SAME       WHEN
BACK        DISCARD    FREQUENCY  OUTPUT   STEP       +<NUMBER>
BREAK       DO          HELP       PAGE     STOP       -<NUMBER>
BYE         END          HISTORY    PERFORM  STOP RUN   SPCFY

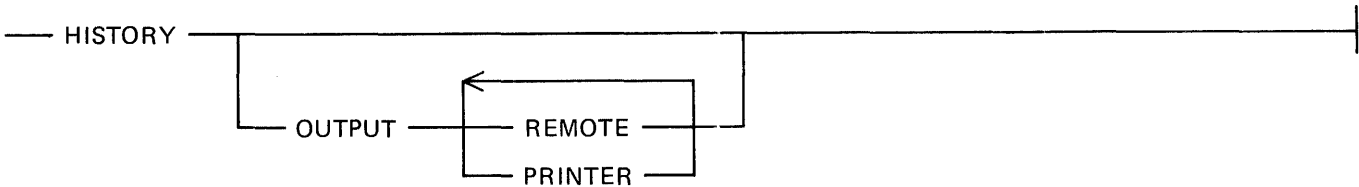
FOR SEMANTICS AND SYNTAX,                ENTER: TEACH <COMMAND>
FOR SYNTAX OF <BRACKET-ITEMS> USED,      ENTER: TEACH CONVENTIONS
TO BEGIN DEBUGGING A COBOL74B/AID PROGRAM, ENTER: EXECUTE <TITLE>
TO CONTINUE PROGRAM EXECUTION,           ENTER: CONTINUE
TO STOP A PROGRAM,                       ENTER: STOP
TO END A PROGRAM EARLY,                   ENTER: STOP RUN
ACCOUNT/PAYROLL ON PACKZ (#565) AT OPEN-FILES
  
```

Figure 4-15. HELP Screen (MO)

HISTORY

The HISTORY command displays the stack entries associated with the current level of PERFORM statements executed in the COBOL74B program. The HISTORY command is only valid during a breakpoint.

Syntax:



Semantics:

OUTPUT

The OUTPUT keyword specifies the output medium for the HISTORY data. If the OUTPUT keyword is omitted, the OUTPUT specification on the screen is used.

REMOTE

Using the REMOTE keyword causes the HISTORY output to be displayed on a remote device.

PRINTER

Using the PRINTER keyword causes the HISTORY data to be printed on a printer device.

Figure 4-16 shows a HISTORY screen.

```

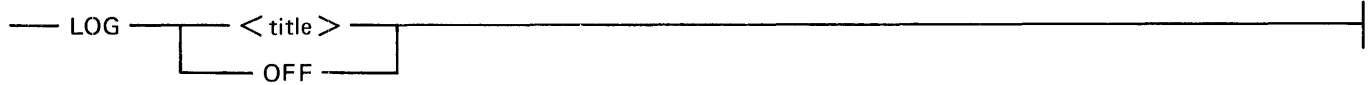
  OUTPUT: REMOTE          FORMAT: EXTERNAL
                          BREAKPOINT TABLE
  1                       : AT CREATE-FIGURES STOP
  2 HEADING               : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP
  3                       : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...
  4                       : AT #0, 9416 STOP
  █                       : HISTORY
  -----
  AT OPEN-FILES HISTORY =
  AT PRINT-DETAIL HISTORY =
  PERFORM THRU K=9, RET.ADR = #0, 15593 IN MPY-WRITE-1
  ACCOUNT/PAYROLL ON PACKZ (#565) AT PRINT-DETAIL
  
```

Figure 4-16. HISTORY Screen (MO)

LOG

The LOG command records subsequent commands in a disk file until a LOG OFF command is entered. The file is activated with the DO command described in this section. This command is valid during all modes of operation.

Syntax:



Semantics:

<title>

The <title> field specifies the name of the LOG file on disk. Refer to Generic Terms in section 1 for a complete description of <title>.

OFF

The OFF keyword terminates recording the COBOL74B/AID commands.

Figure 4-17 shows a LOG screen.

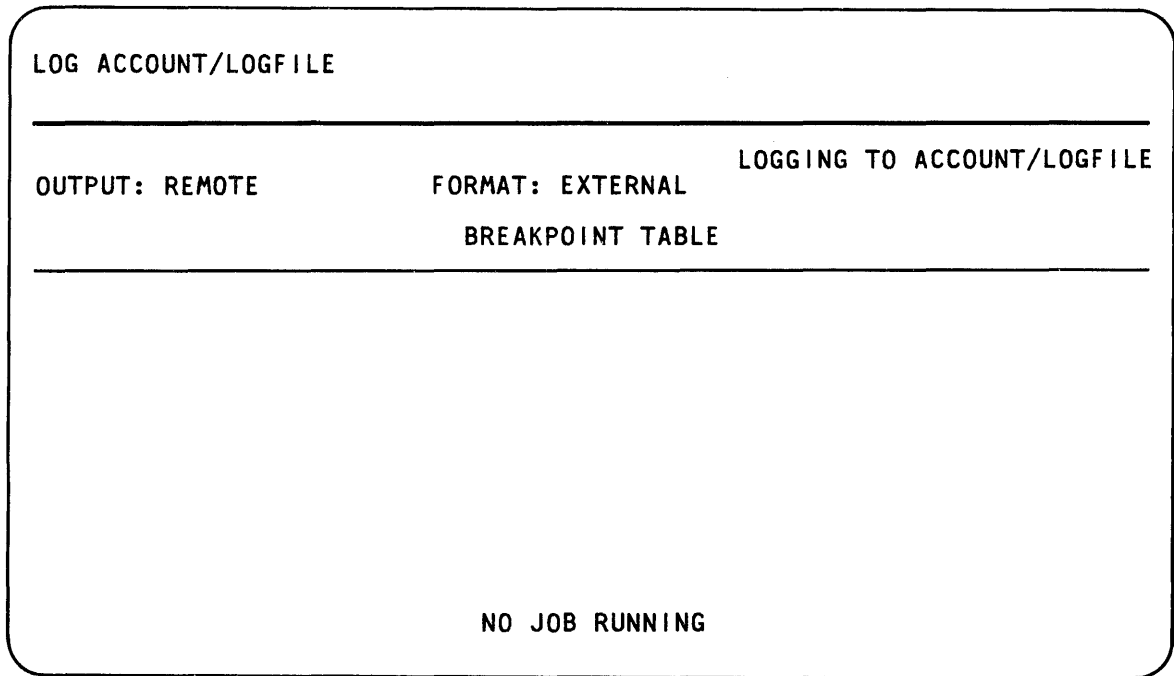
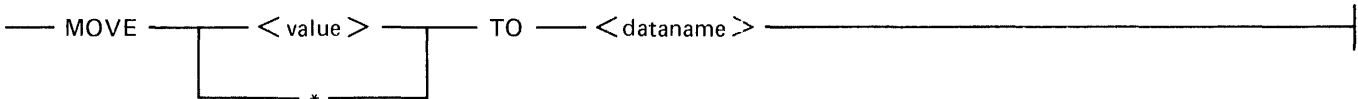


Figure 4-17. LOG Screen (MO)

MOVE

The MOVE command assigns a value to a data item in the current COBOL74B program. This command is similar to the COBOL MOVE statement. The MOVE command can be used independently or in the context of the AT and WHEN commands described in this section. The MOVE command is only valid during a breakpoint.

Syntax:



Semantics:

<value>

The <value> field represents the data to be assigned to the data item in the COBOL74B program. In this case, <value> can be enclosed in parentheses for a literal move. Refer to generic terms in section 1 for a complete description of <value>.

<dataname>

The <dataname> field is the name of a data item in the COBOL74B program. <dataname> cannot exceed 193 characters in length. Refer to generic terms in section 1 for a complete description of <dataname>.

*

The asterisk (*) character forces <dataname> to be reinitialized to the value it had at the beginning of job (BOJ). When this character is used, <dataname> cannot be an array element.

B 1000 Systems COBOL74 Test and Debug System
Installation and Operation Manual
OPERATION

Figure 4-18 shows a MOVE Screen.

```
MOVE 5 TO DAYS-WORKED

-----

OUTPUT: REMOTE          FORMAT: EXTERNAL

                          BREAKPOINT TABLE
1                          : AT CREATE-FIGURES STOP
2 HEADING                 : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP
3                          : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...
4                          : AT 0. 9416 STOP
☐ MOVE 5 TO DAYS-WORKED

-----

NO JOB RUNNING
```

Figure 4-18. MOVE Screen (MO)

NEXT

The NEXT command displays the next page of the trace buffer. This command is only valid in the review mode.

Syntax:

— NEXT —

Figure 4-19 shows a NEXT screen.

```
NEXT
AT PRINT-DETAIL RECORD-COUNT=0
AT MPY-TOTALS RECORD-COUNT=5
AT COMPUTE-STATETAX TAXES=540
AT COMPUTE-FEDTAX TAXES=1235
```

REVIEW OF TRACE BUFFER CLEAR TO EXIT REVIEW MODE

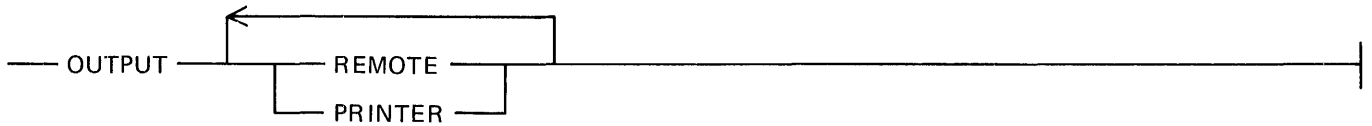
ACCOUNT/PAYROLL ON PACKZ (#565) AT PRINT-DETAIL

Figure 4-19. NEXT Screen (MO)

OUTPUT

The OUTPUT command specifies the output medium. Note the use of the OUTPUT command in the individual commands such as DISPLAY, HISTORY, and TRAIL, described in this section. When the OUTPUT command is specified independently, it becomes the default routing, but when it is specified in the context of other commands, it is only relevant to that particular command. The OUTPUT command is valid during all modes of operation.

Syntax:



Semantics:

REMOTE

Using the REMOTE keyword causes the output generated by the COBOL74B/AID program to be displayed on the terminal.

PRINTER

Using the PRINTER keyword causes the output generated by the COBOL74B/AID program to be printed on a printer device.

Figure 4-20 shows an OUTPUT screen.

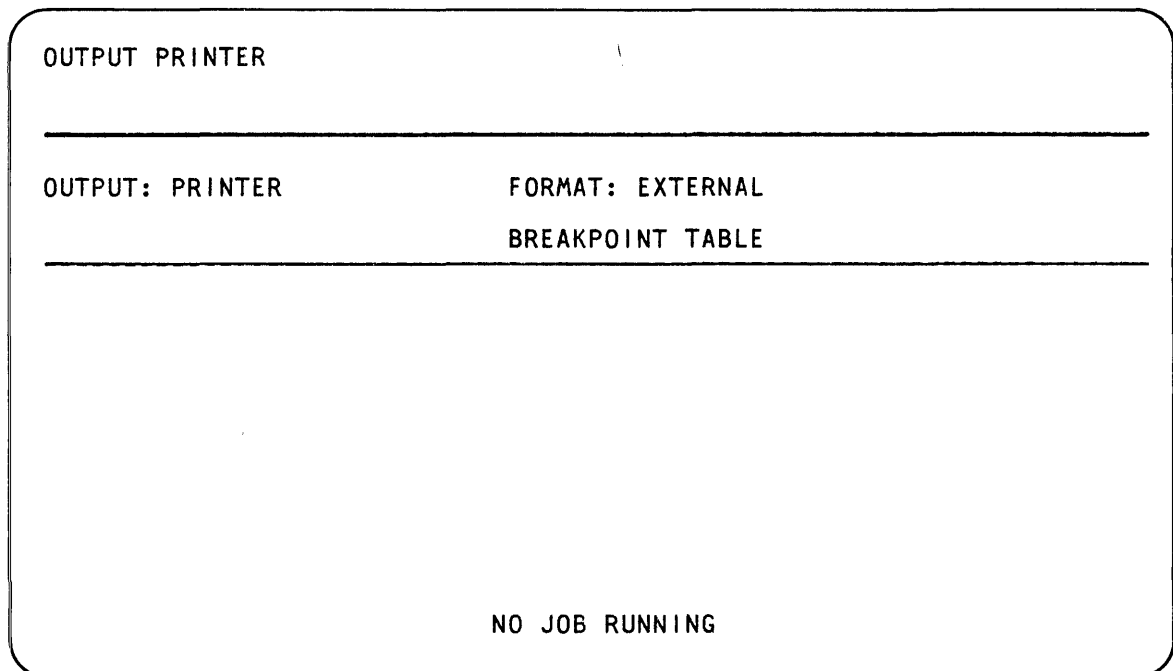


Figure 4-20. OUTPUT Screen (MO)

PRINT

The PRINT command causes either the FREQUENCY, BUFFER, or SCREEN data to be printed. This command is only valid during a breakpoint or when the status is NO JOB RUNNING.

Syntax:



Semantics:

FREQUENCY

The FREQUENCY keyword specifies that the data generated by the FREQUENCY command is to be printed. Refer to FREQUENCY in this section for a complete description of the FREQUENCY command.

BUFFER

The BUFFER keyword specifies that the trace buffer generated by the TRAIL, HISTORY, and DISPLAY (when DISPLAY = REMOTE) commands is to be printed. The maximum that can be printed is the contents of the trace buffer which is 256 lines.

SCREEN

The SCREEN keyword specifies that the screen data above the cursor is to be printed.

Figure 4-22 shows a PRINT screen.

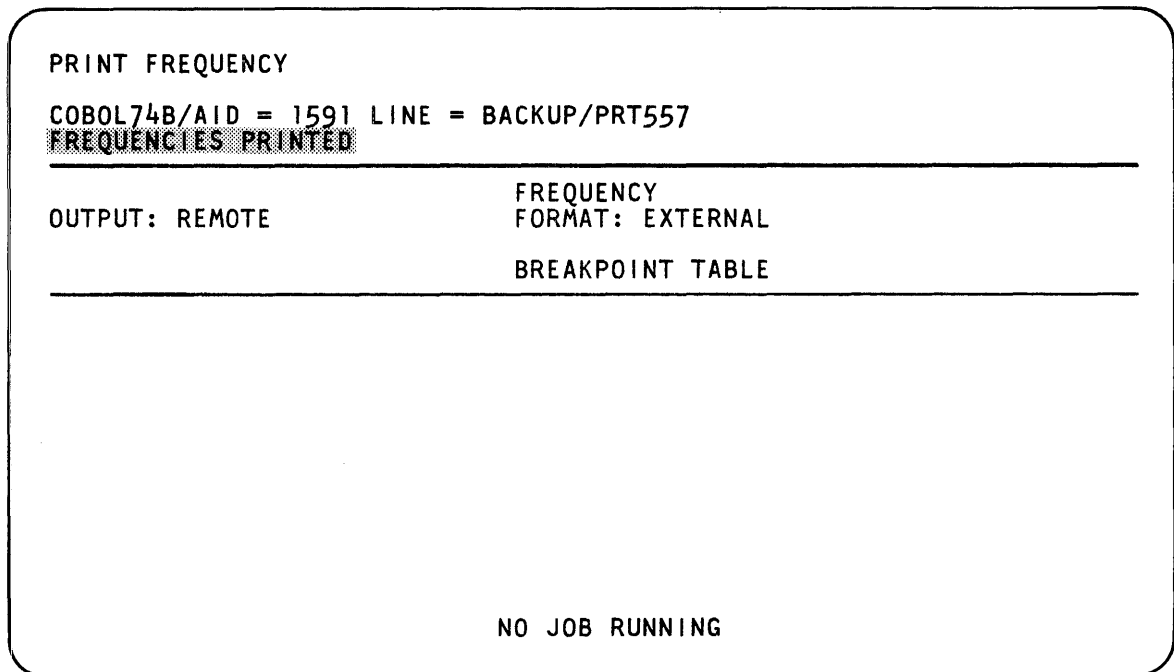


Figure 4-22. PRINT Screen (MO)

REVIEW

The REVIEW command causes the COBOL74B/AID program to enter review mode. Review mode changes the screen format so that the last 22 entries of the trace buffer can be displayed on the screen. Review mode is terminated after the SPCFY key is pressed or the CLEAR command is entered. Refer to the PAGE, NEXT, BACK, and SAME commands described in this section for review mode capabilities. The REVIEW command is only valid during a breakpoint or when the status is NO JOB RUNNING.

Syntax:

— REVIEW —

Figure 4-23 shows a REVIEW screen.

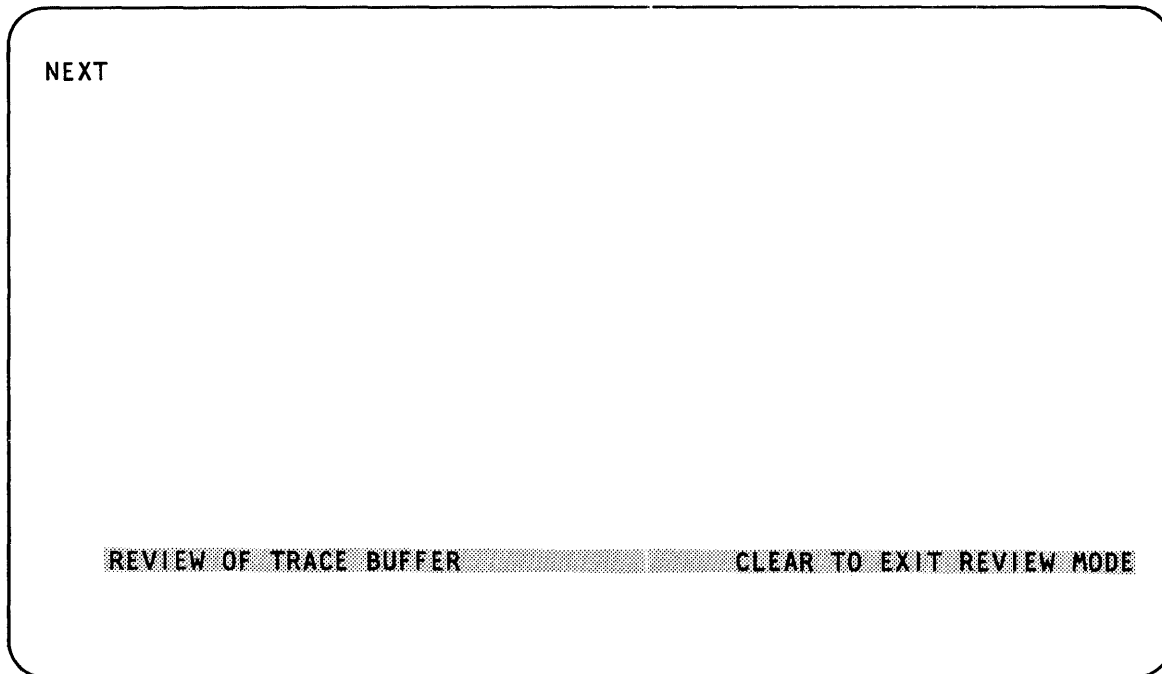


Figure 4-23. REVIEW Screen (R)

SAME

The SAME command displays the current page of the trace buffer on the screen. This command is only valid in review mode.

Syntax:

— SAME —

Figure 4-24 shows a SAME screen.

```
NEXT
AT PRINT-DETAIL REC-CT = 0
AT PRINT-DETAIL REC-CT = 5
AT COMPUTE-TOTALS WORK-HOURS = 40

REVIEW OF TRACE BUFFER CLEAR TO EXIT REVIEW MODE
ACCOUNT/PAYROLL ON PACKZ (#565) AT COMPUTE-TOTALS
```

Figure 4-24. SAME Screen (R)

STEP

The STEP command reinstates the COBOL74B program and then causes it to break at the first paragraph or breakpoint encountered. The break does not become a permanent entry in the breakpoint table. The STEP command is only valid during a breakpoint.

Syntax:

— STEP —

Figure 4-25 shows a STEP screen.

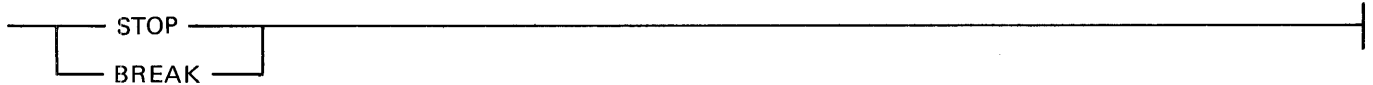
```
STEP
.
-----
OUTPUT: REMOTE          FORMAT: EXTERNAL
                        BREAKPOINT TABLE
1      : AT CREATE-FIGURES STOP
2 HEADING : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP
3      : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...
4      : AT 0, 9416 STOP
-----
                        ACCOUNT/PAYROLL ON PACKZ (#565) AT WRITE-LINE
```

Figure 4-25. STEP Screen (MO)

STOP

The STOP command causes the current COBOL74B program to be suspended. The suspension is a temporary break and does not become a permanent entry in the breakpoint table. This command is only valid when the status is JOB EXECUTING and has no effect on jobs that are already suspended.

Syntax:



Semantics:

BREAK

The BREAK keyword is synonymous with the STOP keyword.

Figure 4-26 shows a STOP screen.

```
STOP  
-----  
OUTPUT: REMOTE          FORMAT: EXTERNAL  
  
                        BREAKPOINT TABLE  
1                       : AT CREATE-FIGURES STOP  
2 HEADING               : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP  
3                       : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...  
4                       : AT 0, 9416 STOP  
█                       : STOP  
-----  
  
ACCOUNT/PAYROLL ON PACKZ (#565) AT PRINT-DETAIL
```

Figure 4-26. STOP Screen (MO)

STOP RUN

The STOP RUN command causes the COBOL74B program to terminate. This command can be issued independently or in context of the AT and WHEN commands described in this section. The STOP RUN command is only valid during a breakpoint.

Syntax:

— STOP RUN —

Figure 4-27 shows a STOP RUN screen. NO JOB RUNNING

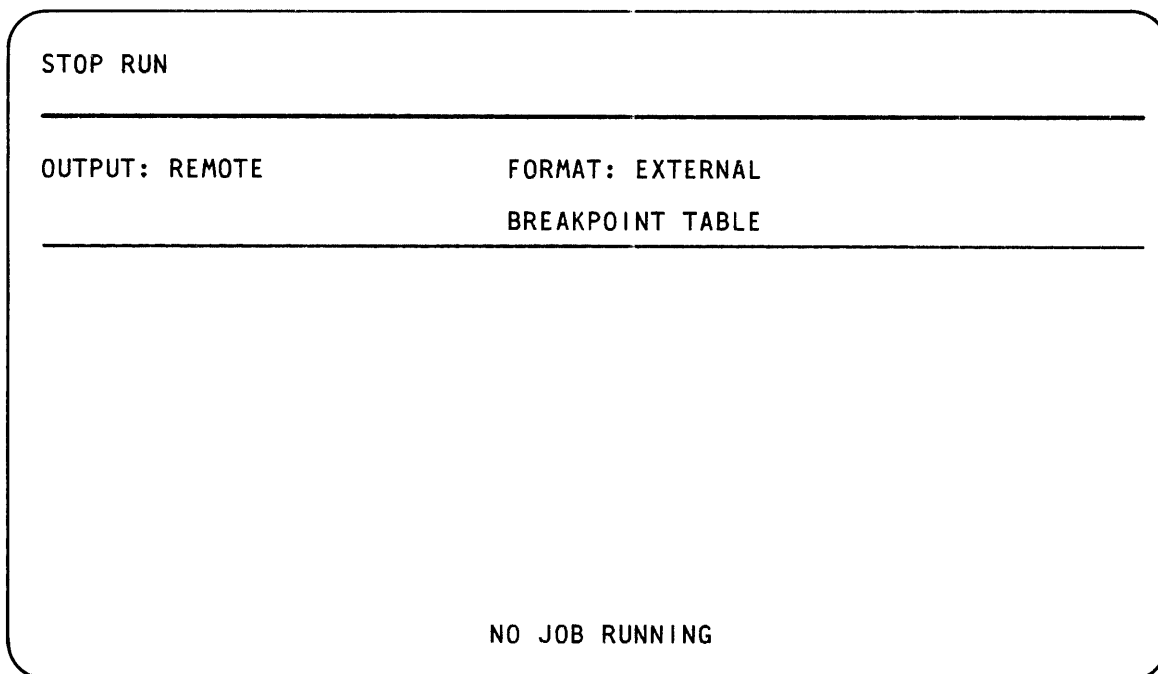
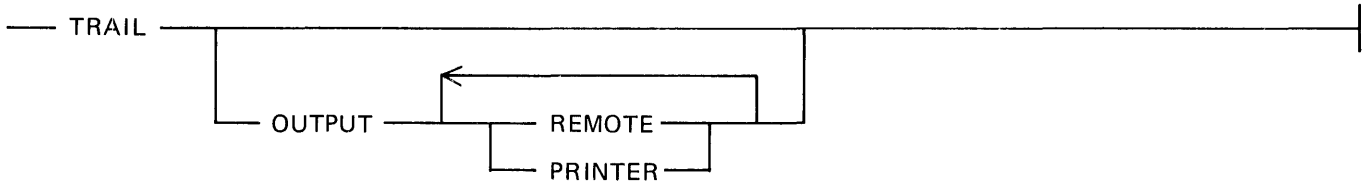


Figure 4-27. STOP RUN Screen (MO)

TRAIL

The TRAIL command causes the COBOL74B/AID program to trace each paragraph encountered in the COBOL74B program and print the results on a printer or terminal. This command is only valid during a breakpoint.

Syntax:



Semantics:

OUTPUT

The OUTPUT keyword specifies the output medium. If the OUTPUT keyword is omitted, the OUTPUT specification on the screen is used.

REMOTE

Using the REMOTE keyword causes the trace data to be displayed on a terminal device.

PRINTER

Using the PRINTER keyword causes the trace data to be printed on a printer device.

Figure 4-28 shows a TRAIL screen.

```

TRAIL
-----
TRAIL TO OUTPUT
OUTPUT: PRINTER                FORMAT: EXTERNAL

                                BREAKPOINT TABLE
1                               : AT CREATE-FIGURES STOP
2 HEADING                       : AT HEAD-ROUTINE WHEN DUMMY-RECORD = CCBS-H-1 STOP
3                               : AT PRINT-DETAIL WHEN TEST-CONTROL MOVE TEST-PERIOD...
4                               : AT 0, 9416 STOP
-----

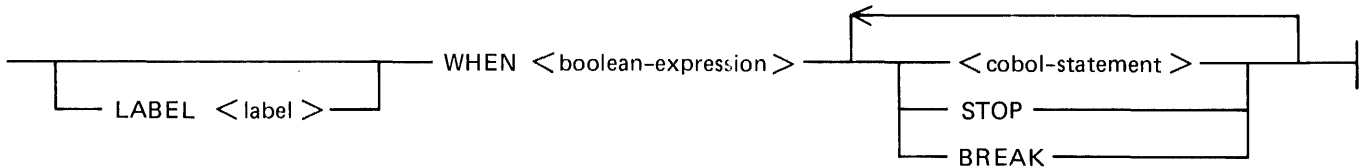
                                ACCOUNT/PAYROLL ON PACKZ (#565) AT PRINT-DETAIL
  
```

Figure 4-28. TRAIL Screen (MO)

WHEN

The WHEN command enters a breakpoint into the breakpoint table. The break occurs only when a specified condition is true. This command can be used in conjunction with the AT command described in this section. The WHEN command is only valid during a breakpoint or when the COBOL74B program is suspended.

Syntax:



Semantics:

LABEL

Using the LABEL keyword assigns a label to the breakpoint.

<label>

The <label> field identifies the name of the label. <label> must be a valid identifier name in the COBOL74B language.

<boolean-expression>

The <boolean-expression> field specifies the conditions for a breakpoint to occur. See Generic Terms in section 1 for a description of the <boolean-expression> field.

<cobol-statement>

The <cobol-statement> field allows the use of the DISPLAY, MOVE, and STOP RUN commands within the WHEN command. See the descriptions of those commands for more information.

STOP

Using the STOP keyword suspends the COBOL74B program at the specified breakpoint. Control is given to the COBOL74B/AID program.

BREAK

The BREAK keyword is synonymous with the STOP keyword.

Figure 4-29 shows a WHEN screen. NO JOB RUNNING

LABEL DAY WHEN DAYS-WORKED = 5 STOP	
<hr/>	
OUTPUT: REMOTE	FORMAT: EXTERNAL
	BREAKPOINT TABLE
DAY	: WHEN DAYS-WORKED = 5 STOP
<hr/>	
NO JOB RUNNING	

Figure 4-29. WHEN Screen (MO)

APPENDIX A

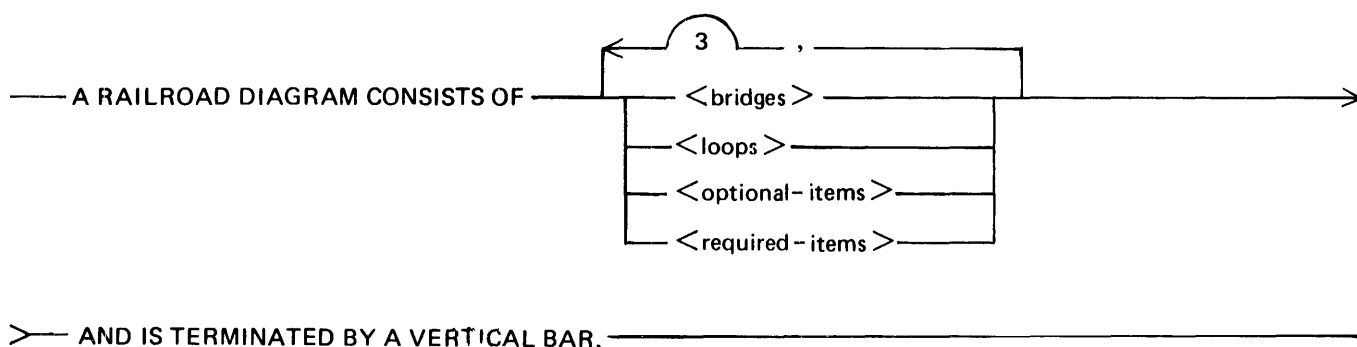
SYNTAX AND NOTATION CONVENTIONS

Railroad diagrams show how syntactically valid statements can be constructed. Traversing a railroad diagram from left to right, or in the direction of the arrowheads, and adhering to the limits illustrated by bridges produces a syntactically valid statement. Continuation from one line of the diagram to another is represented by a right arrow (→) appearing at the end of the current line and the beginning of the next line. The complete syntax diagram is terminated by the vertical bar (|).

Items contained in broken brackets (< >) are syntactic variables that must be further defined or for which the user is required to supply the requested information.

Upper-case items must appear literally. Minimum abbreviations of upper-case items are underlined.

Example:



The following syntactically valid statements can be constructed from the preceding diagram.

A RAILROAD DIAGRAM CONSISTS OF <bridges> AND IS TERMINATED BY A VERTICAL BAR.

A RAILROAD DIAGRAM CONSISTS OF <optional-items> AND IS TERMINATED BY A VERTICAL BAR.

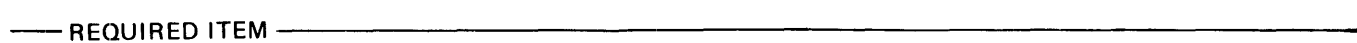
A RAILROAD DIAGRAM CONSISTS OF <bridges>, <loops> AND IS TERMINATED BY A VERTICAL BAR.

A RAILROAD DIAGRAM CONSISTS OF <optional-items>, <required-items>, <bridges>, <loops> AND IS TERMINATED BY A VERTICAL BAR.

Required Items

No alternate path through the railroad diagram exists for required items or required punctuation.

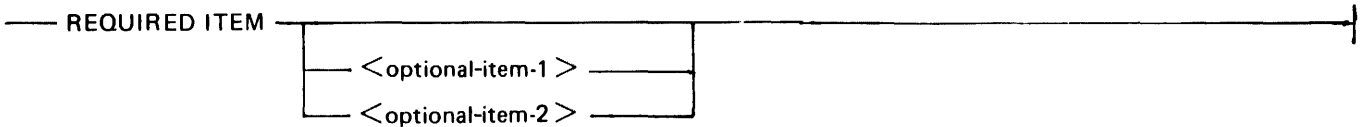
Example:



Optional Items

Items shown as a vertical list indicate that the user must make a choice of the items specified. An empty path through the list allows the optional item to be absent.

Example:



The following valid statements can be constructed from the above diagram.

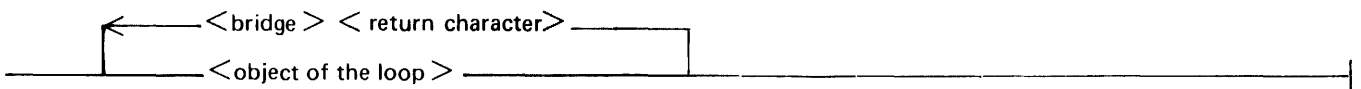
REQUIRED ITEM

REQUIRED ITEM <optional-item-1>

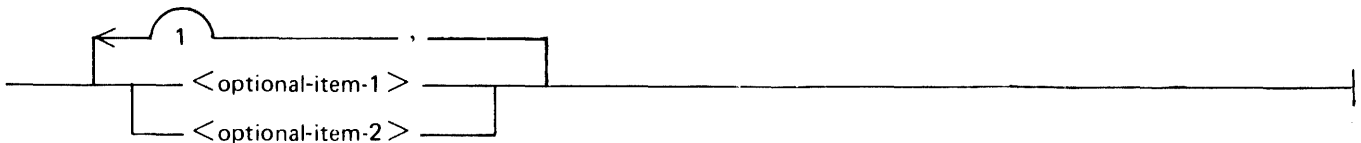
REQUIRED ITEM <optional-item-2>

Loops

A loop is a recurrent path through a railroad diagram. A loop must be traversed in the direction of the arrowheads, and the limits specified by the bridges cannot be exceeded. A loop has the following format:



Example:



The following statements can be constructed from the previous railroad diagram.

<optional-item-1>

<optional-item-2>

<optional-item-1> , <optional-item-1>

<optional-item-1> , <optional-item-2>


<optional-item-2> , <optional-item-1>

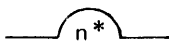
<optional-item-2> , <optional-item-2>

Bridges

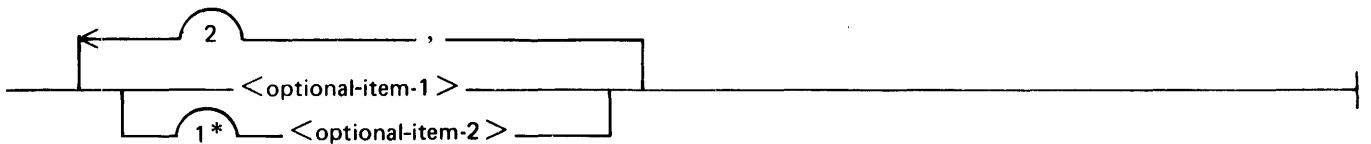
A bridge indicates the minimum or maximum number of times a path can be traversed in a railroad diagram.

There are two forms of bridges.

 n is an integer which specifies the maximum number of times the path can be traversed.

 n is an integer which specifies the minimum number of times the path must be traversed.

Example:



The loop can be traversed a maximum of two times; however, the path for <optional-item-2> must be traversed at least one time.

The following statements can be constructed from the previous railroad diagram.

<optional-item-2>

<optional-item-1>,<optional-item-2>

<optional-item-2>,<optional-item-2>,<optional-item-1>

<optional-item-2>,<optional-item-2>,<optional-item-2>

Keywords

Words appearing in all upper-case letters are keywords and must appear as presented. Words in lower-case letters and embraced in left and right broken bracket characters are symbols for user-supplied words or values. Where keywords can be abbreviated, the abbreviation is denoted by the letters underlined in the keywords. Each keyword, keyword abbreviation, or user-supplied item must be delimited from another such word by at least a single space or by a special character. For example, the parentheses, comma, period, and colon characters can be used as delimiters. Redundant spaces are ignored.

INDEX

* 4-12
ABANDON 4-3
ADOPT 2-2, 4-2, 4-4
asterisk 4-12
AT 4-2, 4-5
BACK 2-4, 4-2, 4-7
boolean-primary 1-2
BREAKPOINT TABLE Area 2-3
breakpoints 2-3
Bridges A-3
BYE 4-1, 4-8
CLEAR 2-2, 4-2, 4-9
COBOL74B/AID Screen Format 2-2
code-address 4-5
command 4-21
COMMAND SET 4-2
condition-name 1-2
constant 1-2
CONTINUE 4-10
dataname 1-2
DISCARD 4-12
DISPLAY 4-14
DO 4-16
Dynamic Memory Setting. 3-1
END 4-1
EXECUTE 2-2, 4-2, 4-17
Executing the COBOL74B/AID Program 4-1
EXIT 4-18
EXTERNAL 4-15
FEW 4-12
FORMAT 4-15, 4-19
FORMAT Field 2-3
FREQUENCY 4-20
General 1-1
Generic Terms 1-2
HELP 4-21
HISTORY 4-22
INTERNAL 4-15
Keywords A-3
label 4-5, 4-12
LAST 4-12
LAST FEW 4-12
LOG 4-23
Loops A-2
main operation screen 2-3
Main Operation Screen Format 2-2
Manual Structure 1-1
MOVE 4-24
NEXT 2-4, 4-2, 4-26

INDEX (CONT)

Operating the COBOL74B/AID Program 4-2
Optional Items A-2
ordinal 4-12
OUTPUT 4-14, 4-22, 4-27
OUTPUT Field 2-3
PAGE 2-4, 4-2
PERFORM 4-28
Plus (+), Minus (-) 4-38
PRINT 4-29
PRINTER 4-14, 4-22, 4-27
procedure-name 1-2
Procedure-name 4-5
Processor Priority 3-1
Related Documents 1-1
REMOTE 4-14, 4-22, 4-27
Remote Files 4-1
Required Files 3-1
Required Items A-1
REVIEW 4-30
Review Screen Format 2-4
SAME 2-4, 4-2, 4-31
screen format 2-2
SMCS Jobs File Entry 3-1
SPCFY 2-2
STEP 4-32
STOP 4-33
STOP RUN 4-1, 4-34
TEACH 4-21
Terminating the COBOL74B/AID Program 4-1
title 1-3
trace buffer 2-4
TRAIL 4-35
value 1-2
welcome screen 2-2
Welcome Screen Format 2-2
WHEN 4-2, 4-36

