

## BRANCH TRACE 2

Emanuel Melichar  
Economist  
Federal Reserve Bank of Richmond  
Richmond 13, Virginia

### PURPOSE

Branch Trace 2 is a branch trace program for IBM 1401 computers with more than 4,000 digits of core storage. A trace program is used in testing and debugging other programs, its output consisting of information about the execution of the program being traced. Branch Trace 2 monitors each instruction of the program being traced and prints one line for each branch that occurs as that program is executed. The contents of this line are as follows:

<u>Print positions</u>	<u>Information printed</u>
1- 3	Address of the instruction at which a branch occurred.
4	Always a blank.
5- 7	Address of the instruction to which the computer branched.
8	Always a record mark.

Thus the output of Branch Trace 2 shows the programmer the sequence of instructions executed in the program being tested, which is of great help to the programmer in cases where it is suspected that the sequence taken was not the one intended. The programmer can then use Minitrace 2 to get further information if necessary.

Uses of Branch Trace 2 are therefore likely to fall into the following categories:

- a) To help debug programs in which core storage limitations prevent the use of Minitrace 2.
- b) To conserve computer time in obtaining a trace in cases where a fault in the sequence of instructions is the probable cause of malfunction of the program. For example, in tests of Richmond program 064, which without tracing ran for 27 seconds, Branch Trace 2 required 1 minute and 55 seconds whereas Minitrace 2 took 8 minutes.
- c) In long programs, to help isolate routines where trouble is present. These can be traced with Minitrace 2, and thus the need for running the entire program against Minitrace 2 is avoided.

### MACHINE REQUIREMENTS

Branch Trace 2 uses 615 digits of core memory and may be assembled anywhere in core except the print area. For example, it may be assembled with origin at 7385 to use positions 7385-7999.

Other machine requirements are:

- More than 4,000 digits of core memory
- Advanced programming feature
  - Index registers
  - Store address register instructions
  - Move record instruction
- 1403 Printer

PROCEDURE

Branch Trace 2 has been designed to avoid almost all "setting-up" of the program deck to be traced. In most cases, it is merely necessary to remove the "END" card from the deck to be traced, place Branch Trace 2 behind the remainder of the deck, place any data cards used behind Branch Trace 2, and load and run the combined deck, following exactly the procedures specified for the program being traced.

SOURCE LANGUAGE

SPS

LIMITATIONS

Branch Trace 2 will trace all generally known 1401 instructions. There are several minor limitations and requirements, fully described in the Write-Up, that are imposed in order to reduce the core space used by Branch Trace 2.

CHECK-OUT STATUS

Branch Trace 2 has been used to trace a variety of programs on systems that include tape, Ramac, and 1407 console equipment. The nature of the program is such, however, that it may still contain bugs or may be unable to handle some instruction or sequence of instructions that was not anticipated by the author. The author therefore requests that each user, as a service to other users, inform him of all difficulties encountered in order that the program may be modified or that a description of the limitation may be added to the Write-Up.

## OPERATING PROCEDURES

### A. Standard trace

1. Remove the last card (END card) from the assembled program to be traced.
2. Place the Branch Trace 2 deck behind the program deck to be traced.
3. Place data cards, if any, behind the Branch Trace 2 deck.
4. Load and run the combined deck.

Follow the procedures specified for the program being traced.  
(Check for switches, carriage control tape, etc.)

Precaution--If not altered, Branch Trace 2 will expect to find the first instruction of the program being traced in core location 333. To begin tracing at a different address, that address should be inserted as the contents of FX, the constant that occupies the first 3 digits of Branch Trace 2. For example, if Branch Trace 1 is assembled with origin at 7385, and a program is to be traced starting at R28, then R28 should be inserted into 7385-7387.

### B. Trace of particular part of program

1. Remove END card from program to be traced.
2. Replace first card of Branch Trace 2 with a card which is identical except that the address at which tracing is to begin is punched into columns 24-26.
3. Put the Branch Trace 2 deck behind the program to be traced.
4. Replace the last card of Branch Trace 2 with the END card from the program to be traced.
5. Place data cards, if any, behind the combined decks.
6. Set address stop to the address at which tracing is to begin.
7. Load and run the program to the address stop. The Start button must be pressed twice at address stops during loading.
8. Set the I-address register to 7388, the address of the first instruction of Branch Trace 2.
9. Press Start to begin tracing.

To stop tracing before the execution of a given instruction, set address stop to the address of the instruction which follows that instruction. A portion or the remainder of the program being traced may then be properly executed by restarting at the address of the given instruction. Tracing may be resumed later in the program by the following procedure:

1. Set address stop to the address of the instruction at which tracing is to be resumed.

2. Press Start and run the program to the address stop.
3. Put the address of the instruction at which tracing is to be resumed into core locations 7385-7387.
4. Reset the I-address register to 7388.
5. Press Start to resume tracing.

#### REQUIREMENTS, LIMITATIONS, AND FURTHER EXPLANATION

##### A. Fully-chained instructions

Branch Trace 2 can accommodate up to 27 full-chain instructions in a row in the circumstances which place the most severe limitation on the number that can be handled. In other words, if the program being traced does not contain a string of more than 27 consecutive fully-chained instructions, it can be traced and the next paragraph can be ignored.

If the chain follows an instruction which is less than 8 digits long, the number of consecutive fully-chained instructions that can be traced is increased by one for each digit by which the instruction preceding the chain is shorter than 8. If the chain is not followed by a 4-digit instruction whose op-code is either M, L, Q, or H, or by a 4-digit constant whose first character is M, L, Q, or H, the number of consecutive fully-chained instructions that can be traced is increased by 4.

##### B. Branch Trace 2 sets a word mark in 001

A word mark is set in core location 001 and remains set during execution of the program being traced. The latter must therefore be able to function under this condition.

##### C. Stacker instructions

A Select Stacker instruction given after a Read instruction will not be effective because it will not be executed within the necessary time limit. All cards read should be expected to fall into the normal read pocket.

##### D. Word marks must follow all instructions

It is recommended that programs that may be traced be written with word marks in the location following each instruction, thus extending the general requirement to the three instructions that do not ordinarily require such a word mark (the 4-digit unconditional Branch, the 7-digit Set Word Mark, and the 7-digit Clear Storage and Branch).

However, Branch Trace 2 will usually be able to execute the above instructions if a word mark occurs in core within 32 digits after the last digit of the instruction (within 35 digits of the 4-digit unconditional Branch).

The author's experience has shown that this requirement must especially be kept in mind when writing the last instruction of the program (the instruction preceding the END card) and when patching assembled decks (also remember not to put a patch into locations used by Branch Trace 2).

E. Partial logic of Branch Trace 2

The following statement of the principal logic employed by Branch Trace 2 may be useful in determining whether programs which make unusual use of particular instructions can be traced.

1. The contents of the B-address register after execution of an instruction being traced are stored and returned to the register before execution of the next instruction if the latter is a 4-digit Move, Load, or Store B-address Register instruction. The contents of the A-address register are similarly handled if the next instruction to be traced is a 4-digit Store A-address Register instruction. These are the only cases in which the contents of the registers are stored for use by the next instruction. When a Branch occurs, the address of the next sequential instruction following the Branch instruction is introduced into the B-address register prior to execution of the next instruction to be traced.
2. Branch Trace 2 recognizes the following 1-digit instructions as fully-chained instructions and causes them to be executed without tracing at the same time that the preceding unchained instruction is executed and traced:

C	□
X	!
W	?
%	L
Q	M
H	/
Z	#
@	E
P	V
D	B
Y	S
,	A

3. Branch Trace 2 recognizes that branches may occur to the A-address of instructions with the following op-codes. A trace is printed if the branch does occur.

B	3
V	5
W	6
.	7
1	K
4	F
2	/ (7-digit instruction only).

Sample Output of Branch Trace 2

Trace of Richmond Program 064  
 Inquiry into account of Transit Department  
 on Ramac, using 1407 Console

363 347\*

363 347\*

358 530\*

557 501\*

500 371\*

390 434\*

412 478\*

497 561\*

573 738\*

218 TRANSIT

TOTAL DEBITS	87,415,879.13
TOTAL CREDITS	22,318,508.89

	DEBITS	CREDITS
--	--------	---------

874 473\*

494 503\*            27.50

27.50

947 862\*

6,321.24

947 862\*

7.13

947 862\*

1,050.00

947 862\*

374 473\*

494 903\*            70,000.00

70,000.00

947 862\*

494 473\*

494 903\*            2,776,362.21

2,776,362.21

947 862\*

PG LIN CT LABEL OP A OPERAND B OPERAND D LOC ERROR NOTES COMMENTS BRTR:

PG	LIN	CT	LABEL	OP	A OPERAND	B OPERAND	D	LOC	ERROR	NOTES	COMMENTS	BRTR:
					NO CONTROL CARD							
1	030	3	FX	DCW	*							333
1	040	1	BFAD	DCW	*							
1	050	3	191	DCW	*							191
1	060	1	TOP	DCW	*							
1	070	1	EXCUTE	DCW	*							N
1	080	3	AREG	DC	*							000
1	090	3	BREG	DC	*							000
1	100	1	A	DCW	*							
1	110	1	B	DC	*							
1	120	3	C	DC	*							
1	130	3	F8X	DC	*							
1	140	1	D	DC	*							
1	150	4		DC	*							
1	160	30	ENDEX	DC	*							
1	170	4	E2	CS	ENDEX							
1	180	4		SW	0001							
1	190	7		MCH	FX	TA				£002		
1	200	7		MCH	0089	INDEX1						
1	210	7		MCH	0094	INDEX2						
1	220	7		MCH	BLANKS£002	0089						
1	230	7		MCH	FX	0094						
1	240	7	E10	SBR	0094	0001	2					
1	250	8		BWZ	E15	0000	2					1
1	260	7		SBR	0089	0001	1					
1	270	4		B	E10							
1	280	7	E15	MCH	0000	2	B					1
1	290	1		LCA								
1	300	7		MCH	BLANKS	B	1					
1	310	7		MCH	F8X -004	AFAD						
1	320	7		MCH	F8X -001	BFAD						
1	330	7	E22A	MCH	0000	2	TOP					
1	340	7		SBR	0094	0001	2					
1	350	8		BWZ	TEST	0000	2					1
1	360	7	WH	MA	191	0094						
1	370	7	ZZ	SBR	0094	0004	2					

PG LIN CT LABEL OP A OPERAND B OPERAND C D LOC ERROR NOTES COMMENTS BRTR

1	380	8	BWZ	MOD2		0000	2	I				
1	390	7	E26AA	MA	:19F	0094						
1	400	4		E26								
1	410	8	MOD2	B	E24	TOP		M				
1	420	8		E24		TOP		L				
1	430	8		E24		TOP		H				
1	440	8		E22A45		TOP		Q				
1	450	4		E26AA								
1	460	7	E24	LCA	HXXX &003	C		I				
1	470	7	E26AB	SBR	0089	0004		I				
1	480	4		E26AA								
1	490	7	E26	LCA	BXXX &003	C		I				
1	500	7		MCW	0094	FX						
1	510	7		MCW	INDEX1	0089						
1	520	7		MCW	INDEX2	0094						
1	530	8		B	EXCUTE	AFAD						
1	540	7		SBR	TXYL &003	TABLE1						
1	550	7	TXYL	MCW	TABLE1	LOOK2 &007						
1	560	4		SAR	TXYL &003							
1	570	8	LOOK2	B	E60	A		F				
1	580	8		B	TEST2	LOOK2 &007		B				
1	590	4		B	TXYL							
1	600	1		DCW	*						B	
1	610	1		DCW	*						V	
1	620	1		DCW	*						W	
1	630	1		DCW	*						.	
1	640	1		DCW	*						1	
1	650	1		DCW	*						4	
1	660	1		DCW	*						2	
1	670	1		DCW	*						3	
1	680	1		DCW	*						5	
1	690	1		DCW	*						6	
1	700	1		DCW	*						7	
1	710	1		DCW	*						K	
1	720	1	TABLE1	DCW	*						F	
1	730	8	TEST2	B	E59	A		/				

PG	LN	CT	LABEL	OP	A OPERAND	B OPERAND	D	LOC	ERROR NOTES	COMMENTS	BRTR
1	740	4		B	EXECUTE						
1	750	7	E62	MCW	FX	BREG					
1	760	7		MCW	AFAD	FX					
1	770	7	COMP	MCW	0208	STORE2					
1	780	7		MCW	RM	0208					
1	790	7		MCM	0201	WR					
1	800	7		MCW	0089	INDEX1					
1	810	7		SBR	0089	0201					
1	820	7	CHAR	MCM	TA	-201 1	1				
1	830	4		SBR	0089						
1	840	8		B	DONE3	0089	9				
1	850	4		B	CHAR						
1	860	2	DONE3	CC			S				
1	870	3		W							
1	880	7		MCM	WR	0201					
1	890	7		MCW	STORE2	0208					
1	900	7		MCW	INDEX1	0089					
1	910	4		B	E2						
1	920	1	WR	DCW	*						
1	930	6		DC	*						
1	940	1	RM	DCW	*						
1	950	1	BLANKS	DCW	*						
1	960	2		DC	*						
1	970	3	INDEX1	DCW	*						
1	980	3	INDEX2	DCW	*						
1	990	4	QXXX	SAR	AREG						
2	000	1	STORE2	DCW	*						
2	010	1	TA	DCW	*						
2	020	3		DC	*						
2	030	3	AFAD	DCW	*						
2	040	1		DCW	*						
2	050	7	TEST	SBR	TXY	8003					
2	060	7	TXY	MCM	TABLE	LOOK1	8007				
2	070	4		SAR	TXY	8003					
2	080	8	LOOK1	B	E22A34	TOP	A				
2	090	8		B	WV	LOOK1	8007				



BRANCH TRACE 2

*Added requirement?  
(See below)*

Emanuel Melichar  
Economist  
Federal Reserve Bank of Richmond  
Richmond 13, Virginia

PURPOSE

Branch Trace 2 is a branch trace program for IBM 1401 computers with more than 4,000 digits of core storage. A trace program is used in testing and debugging other programs, its output consisting of information about the execution of the program being traced. Branch Trace 2 monitors each instruction of the program being traced and prints one line for each branch that occurs as that program is executed. The contents of this line are as follows:

<u>Print positions</u>	<u>Information printed</u>
1- 3	Address of the instruction at which a branch occurred.
4	Always a blank.
5- 7	Address of the instruction to which the computer branched.
8	Always a record mark.

Thus the output of Branch Trace 2 shows the programmer the sequence of instructions executed in the program being tested, which is of great help to the programmer in cases where it is suspected that the sequence taken was not the one intended. The programmer can then use Minitrace 2 to get further information if necessary.

Uses of Branch Trace 2 are therefore likely to fall into the following categories:

- a) To help debug programs in which core storage limitations prevent the use of Minitrace 2.
- b) To conserve computer time in obtaining a trace in cases where a fault in the sequence of instructions is the probable cause of malfunction of the program. For example, in tests of Richmond program 064, which without tracing ran for 27 seconds, Branch Trace 2 required 1 minute and 55 seconds whereas Minitrace 2 took 8 minutes.
- c) In long programs, to help isolate routines where trouble is present. These can be traced with Minitrace 2, and thus the need for running the entire program against Minitrace 2 is avoided.

MACHINE REQUIREMENTS

Branch Trace 2 uses 615 digits of core memory, ~~and may be assembled anywhere in core except the print area.~~ For example, it may be assembled with origin at 7385 to use positions 7385-7999.

*It must be assembled to begin in a location ending in 85.*

Other machine requirements are:

- More than 4,000 digits of core memory
- Advanced programming feature
- Index registers
- Store address register instructions
- Move record instruction
- 1403 Printer