

BELLCOMM, INC.

SUBJECT: AUTODIAGRAM Version for UNIVAC 1108-  
Case 803

DATE: December 15, 1967

FROM: Mrs. G. I. Hain

ABSTRACT

The flowcharting program AUTODIAGRAM is now running on the UNIVAC 1108. AUTODIAGRAM draws flowcharts directly from FORTRAN source code, and/or UNIVAC assembly language source code, and/or PERT or PERT-like data cards. This memorandum describes the features of AUTODIAGRAM available in the UNIVAC 1108 version, and gives examples for typical RUN setups.

GPO PRICE \$ \_\_\_\_\_

CFSTI PRICE(S) \$ \_\_\_\_\_

Hard copy (HC) 3.00

Microfiche (MF) .65

ff 653 July 65

N 6 8 - 1 6 2 3 8

FACILITY FORM 502

(ACCESSION NUMBER)	14	(THRU)	
(PAGES)	CP 92508	(CODE)	08
(NASA CR OR TMX OR AD NUMBER)		(CATEGORY)	

# BELLCOMM, INC.

---

SUBJECT: AUTODIAGRAM Version for UNIVAC 1108 - DATE: December 15, 1967  
Case 803

FROM: Mrs. G. I. Hain

## MEMORANDUM FOR FILE

### INTRODUCTION

The flowcharting program AUTODIAGRAM is now available for the UNIVAC 1108. AUTODIAGRAM displays the logical structure of computer programs directly from the FORTRAN source code, and/or assembly language source code, and/or PERT networks from PERT or PERT-like data cards. The resulting flowchart is a directed graph consisting of elements (boxes) and their connections (lines). Loops are represented by repeating the loop entrance element at the end of the loop as "loop end element".

The part of AUTODIAGRAM which arranges the elements and sets up their graphical display is the same as described in 1,2,3,4. It is converted from the IBM 7040 and part of it from the IBM 360 version. The source input scan has been altered to expand the FORTRAN scan to FORTRAN V, to include the UNIVAC assembly language and to read CUR files.

This memorandum describes the features of AUTODIAGRAM and gives examples for typical RUN setups. This is followed by a description of the option parameters and by a description of the scanning process for the different input types together with their specific options.

- 
1. G. Hain, K. Hain, "An Automatic Flowchart Design", Proc. of ACM, Cleveland, Ohio, 1965.
  2. G. Hain, K. Hain, "An AUTO-DIAGRAMMER", NASA Technical Memorandum, NASA TM X-1317.
  3. G. Hain, "A BCMSYS Program for Generating Flowcharts" - Bellcomm - Memorandum for File, Case 803, July 12, 1967.
  4. K. Hain, "AUTODIAGRAM for System 360", NASA Goddard Space Flight Center, Greenbelt, Md., May, 1967.

AUTODIAGRAM FEATURES

AUTODIAGRAM draws flowcharts from a source deck(s), which can be a file in CUR format, or a card deck, and is one of the following types:

- FRTRA - FORTRAN IV or FORTRAN V source decks
- UNIV - UNIVAC assembly language
- PERT5 - Normal PERT data cards
- PERT7 - NASA PERT data cards
- LOGIC - Element names and connection matrix given on data cards

The result of the source deck analysis consists of the listing of the input, the cross-reference table and the flowchart depending on the output option:

- No specification - flowchart on the line printer
- CUBIC - flowchart on the SC-4020 plotter, rectangular connection lines
- PLOT - flowchart on the SC-4020 plotter, diagonal connection lines
- PRINT - Only the above mentioned listing and the cross-reference table, no flowchart

No further specification is needed to draw a flowchart. But AUTODIAGRAM has features which are especially useful for larger programs, say more than 200 elements, and for those with a complicated loop structure (many nested loops). For such programs AUTODIAGRAM has the additional abilities to display selected parts of the flowchart and/or to draw a contracted flowchart by setting one of the modification parameters.

- LOOP - Contract each loop to one element and display the contents of the loop as a higher level flowchart part. Nested loops are displayed in a hierarchical order, the innermost loop has the highest level.
- BREK=n - AUTODIAGRAM splits the flowchart into subparts - not exceeding approximately n elements - and sets the proper interfaces.
- NLOG - AUTODIAGRAM splits the flowchart into subparts. The beginning and end of such a subpart have to be specified by control cards. Those subparts can be displayed in detail and/or contracted to one operator element within the remainder of the flowchart.

The control information for AUTODIAGRAM is given on three parameter cards. They are:

1. FLOW\$ - separator and special option card
2. A title card - identifying the flowchart
3. Input type and output option card

AUTODIAGRAM runs as a user program under EXEC II. It can handle up to 1000 elements and up to 3000 connections between those elements. The estimated run time consists of CUR operations involved (if any), loading AUTODIAGRAM (30 to 40 secs.), and the flowcharting process of the source deck which depends on size, options and logic of the deck, and needs 2 to 4 seconds per inch of the source deck.

#### RUN SETUP

AUTODIAGRAM is stored on FASTRAND under \$GIH.A; the name of the MAP element is FLOW. The first three parameter cards for AUTODIAGRAM must directly follow the @ XQT FLOW card. If the input comes from the card reader, all source decks will be flowcharted under the supervision of those three cards until the next three parameter cards are read. If the input comes from a CUR file, all decks up to the END of FILE of the CUR file are flowcharted. After the END of FILE, the card reader is inspected for further parameter cards.

If the flowchart is to be drawn by the SC-4020, a tape must be assigned to UNIT J (logical 10) with an assigned card. The output is directly acceptable to the SC-4020. The flowcharts for different decks (or subparts) are separated by END of FILE marks. The convention is to assign J=PLOT together with a MSG card to save this tape for plot output. The operator will then return the number of the scratch tape, and the user must then deliver the plot request to the dispatcher. (The number of files and the number of frames are part of the print output.)

AUTODIAGRAM may use the logical file numbers 15, 16, 17, 18, 19 and 20 (File O, P, Q, R, S, and T under NTAB\$/BCM) as scratch files. If other work is being done under the same RUN card, those files should be used only as scratch files.

Typical RUN setups are demonstrated in the following examples. A complete list of the options is given below.

1. In the first example only the input type is specified - FRTRA (FORTRAN source deck(s)). The deck(s) are read from the card reader, no unit specification is needed. The resulting flowchart(s) is printed, no output specification is needed. The following control card setup flowcharts a group of FORTRAN source decks.

```

.      PUN      FLOW1,EX8GIH,2,75
.      HDG      G.I. HAIN - CARD READER INPUT EXAMPLE
.      ASG      A=$GIH.A          . AUTODIAGRAM PROGRAM FILE
.      XQT      CUR
.      ERS
.      IN       A
. N     XQT      FLOW          . ELEMENT NAME OF PROC. MAP
FLOW$
      INPUT FROM CARD READER, FLOW CHART OUTPUT ON THE PRINTER
FRTRA
C          ONE OR MORE FORTRAN DECK(S)
. FIN

```

NOTE: The card @ FOR NAME,NAME is not part of the FORTRAN source deck, but a call for the EXEC II FORTRAN compiler, and must be removed. Any @ in column 1 on a card terminates the program in execution.

2. In the next example, one or more FORTRAN decks are submitted as a CUR file on tape 1884. Under the same request AUTODIAGRAM will flowchart all routines up to the END of FILE of the CUR file. After reading the EOF, the card reader is checked for another set of control cards.

In this example the tape is assigned to B. Then "2", the logical number of B, followed by "," must be given as a UNIT on the flowchart control card 1. On the same card the option "LOOP" is specified, the resulting flowchart will display the loops in a hierarchical order.

The output option CUBIC is punched on parameter card 3, so that the flowchart will be on the SC-4020. In this case a tape must be assigned to file J.

```

'   RUN      FLOW2,EX8GIH,2,75
'   HDG      G.I. HAIN - FORTRAN DECKS AS A CUR FILE ON TAPE 1884
'   ASG      B=1884
'   MSG      PLEASE SAVE TAPE PLOT AND NOTE THE NUMBER FOR ME
'   ASG      J=PLOT           . PLOT OUTPUT SCRATCH
'   ASG      A=$GIH.A         . AUTODIAGRAM PROGRAM FILE
'   XQT      CUR
'   ERS
'   TRW      J,B
'   IN       A
'N   XQT      FLOW           . ELEMENT NAME OF PROC. MAP
FLOW$ UNIT=2, LOOP=,
DISPLAY THE LOOPS IN HIERARCHICAL ORDER, PLOT OUTPUT ON THE SC 4020
FRTRA CUBIC
' FIN

```

3. In the third example three assembly decks are to be flowcharted from a CUR file. The flowchart for the first two is to be printed (no output option is specified). The flowchart for the third routine is to be drawn by the SC-4020 (CUBIC is punched on parameter card 3 of the second set). Two sets of control cards are required. The selected routines are written on a scratch file (D), and the third routine is separated by an EOF.

Deck 1 and deck 2 are plotted by the first set of flowchart control cards.

After reading the EOF, the next control card set is read from the card reader, and deck 3 is plotted. The chosen location counter is "1", punched as LOCT=1, on control card one.

```

'   RUN      FLOW3,EX8GIH,2,75
'   HDG      G.I. HAIN - SELECT ASSEMBLY DECKS FROM A CUR FILE
'   ASG      A=$GIH.A         . AUTODIAGRAM PROGRAM FILE
'   ASG      B=1800
'   MSG      PLEASE SAVE TAPE PLOT AND NOTE THE NUMBER FOR ME
'   ASG      J=PLOT           . PLOT OUTPUT SCRATCH
'   XQT      CUR
'   ERS
'   IN       B
'   TWR      D,CONVRT
'   TWR      D,CHANGE
'   TEF      D
'   TWR      D,LAST
'   TEF      D
'   TRW      D
'   ERS
'   IN       A
'N   XQT      FLOW           . ELEMENT NAME OF PROC. MAP
FLOW$ LOCT=1, UNIT=4,
CONVRT AND CHANGE ARE STORE IN THE SAME FILE, FLOW CHART ON THE PRINTER
UNIV
FLOW$ UNIT=4, LOCT=1,
LAST IS IN THE SECOND FILE, FLOW CHART ON TAPE FOR PLOT. BY SC4020
UNIV CUBIC
' FIN

```

PARAMETER CARDS AND OPTIONS

For most programs the specification of the input type is sufficient. Eventually the user will request for loop analysis (LOOP), or choose one of the output options, or specify an input unit, or control the location counter of assembly programs. This is generally true for programs which consist of less than 200 to 300 elements. But for large or logically complicated programs the special split abilities of AUTODIAGRAM are desirable.

The easiest way to split a large program into subparts is to set BREK=200,. AUTODIAGRAM sets up the whole logical flow structure and splits it into subparts of less than 200 elements and sets the interfaces.

For specific applications (i.e., searching a connection error, displaying a special part of the logic) it may be desirable to flowchart only (and/or suppress) certain parts of the flowchart. This can be achieved by plotting first the whole flowchart and then selecting the proper limits for the subparts from the flowchart. The lower limit must be connected in a forward direction with the upper limit, otherwise AUTODIAGRAM will give an error message and will not eliminate the subpart. The name of the lower limit element (for FORTRAN printed under the box, otherwise printed above the box) gives the first element name on the KEY card. The name of the upper limit element gives the second name for the KEY card. NLOG=1, display only this part of the flowchart. NLOG=2, contract this part to one element in the flowchart.

The split options are applicable for any size of the source deck inside the table limits. If the element table or the connection matrix overflows, the program must be divided by "hand" into smaller decks before it can be plotted. The choice of the options depends on the structure of the program and the preference of the user. If the option LOOP is specified together with BREK and/or NLOG, the options BREK, NLOG are ignored, because AUTODIAGRAM needs the coordinates as they are printed on the cross-reference table to find the proper place to display the contents of the loop.

For type UNIV the option SKIP is used by the input scan and can prevent table overflow (i.e., can be used to skip large EQU tables, data definitions, etc.).

In the following all options and their format are described:

CARD 1 - has the control card mark FLOW\$ and can have the following options:

Col. 2-6      Col. 7-72

FLOW\$          UNIT=n, LOCT=n, SKIP=n, LOOP=, BREK=n, NLOG=n,  
(n can be any positive integer consisting of 1 to 3 digits.)

The options can be given in any order, with the restriction that no blanks are embedded inside a field and that the integer must be followed by ", ".

Default option is: The source input comes from the card reader. Flowchart the whole source deck as one deck without modifications. For type UNIV only, use location counter zero and start scanning with the first source deck card. The standard set-up is changed by specifying the following options:

- UNIT=n, - The source deck is to be read from UNIT n.
- LOCT=n, - Plot only those assembly language statements which belong to location counter n, and/or are counter-independent (for type UNIV only).
- SKIP=n, - Skip n cards before start of scanning of the source deck (for type UNIV only).
- LOOP=, - Contract each DO loop and each logical loop to one element and display the contents of the loop as a higher level subpart. Connect the contracted loop by a specific interface sign with this subpart.
- BREK=n, - The flowchart should be split into subparts consisting of no more than n elements per subpart, with interfaces to the remainder.
- NLOG=n, - Split the flowchart in subparts (not to be used for PERT7). The ranges of the subparts are given on special KEY control cards (at least one) which will follow the title card.

NLOG=1, give separate nets for each KEY control card.



NLOG=2, display one contracted net (omit all strings given by KEY cards)

NLOG=16, plot the flowchart of the whole net.

These features are additive, e.g., NLOG=19, gives the flowchart of the whole net, a flowchart of the contracted net, and for each KEY card a subnetwork.

This option is only valid for the deck following the control card set. After the first deck has been plotted, NLOG is reset to blank.

- NOTE: a. The order of elements in the n=declaration may be slightly different from the order printed in the total cross-reference table but only inside one level (x=coordinates).
- b. If NLOG=2, is given and ranges of KEY cards overlap, this can result in unexpected connections.

KEY control card specification:

Col. 1-2 - must be one of the options KK, KN, NK, NN.

Col. 8-12 - lower limit name (nameL)

Col. 14-18 - upper limit name (nameU)

KK - select the logical product of all forward connections with nameL and all backward connections of nameU (with interface to the remainder).

KN - select all elements connected forward with nameL up to nameU.

NK - select all elements connected backward from nameU up to nameL.

NN - select all elements between the limits nameL and nameU.

CARD 2 - Col. 1-72 any alphanumeric information for the title.

CARD 3 - is the type and output option card.

Col. 1-6 must be one of the type options:

= FRTRA, = PERT5, = PERT7, = LOGIC, = UNIV,

- Col. 7-12 = blank, use the printer to plot the flowchart.
- = PLOT, store the flowchart on a SC-4020 tape, use for the connection lines the shortest distance (diagonal connection).
  - = CUBIC, store the flowchart on an SC-4020 tape, draw the connection lines in horizontal and vertical directions.
  - = PRINT, print only the listings.
- Col. 13-18 = blank, the source deck starts with the first card.
- = TEXT - all cards up to the delimiter card Col. 2-6 \*\*\*\*\* are not part of the flowchart, but will be printed ahead of the plotted output as description.

NOTE: If TEXT is given, it is used only for first deck or subpart of a deck, and then reset to blank.

#### SOURCE INPUT

FORTRAN and UNIVAC assembly languages are scanned for transfer instruction in order to result in a logical connection matrix. PERT5, PERT7 and LOGIC give this matrix directly and only double connections are eliminated.

#### Type - FRTRA

FORTRAN IV, FORTRAN V source decks. For FORTRAN each element includes one or more statements. Each statement which can be reached by a transfer is the starting point of an element. Every branch statement ends an element. In addition every statement number, every call statement and every I/O statement is regarded as a starting point for an element. Besides the statement number, each element is characterized by a statement counter, which is used by AUTODIAGRAM as a name for the element. The END statement or the FUNCTION or SUBROUTINE statement of the next routine is used as an END of INPUT for this deck.

#### Type - UNIV

UNIVAC assembly language deck. The user can specify the location counter by punching LOCT=n, on the delimiter card. Only those instructions and directives which are location counter independent and/or follow the selected location counter are used for the flowchart and printed on the input list. The card counter is incremented for all cards read.

An element includes one or more instructions. An element is started if it follows a data block or is reached by any type of transfer. Any type of transfer instruction ends an element. In addition an element can be started if:

- a. It is not in the list of assembly instructions or directives.
- b. It is one of the instructions:  
EX, HJ, HKJ, ER, TS or the directive .b\*\*.  
(b = BLANK)  
(The interrupt location for ER is given the name \$SYER and TS the name \$SYTS.)
- c. The transfer to a subroutine is marked with an indirect bit or followed by an index register.

In all such cases the "CODE" is marked with "\*". "DO" definitions are always taken as true, and the task of the "DO" is always accepted. The card scan starts after the "b," delimiter of the DO directive.

PROC and FUNC definitions are skipped as were the MACRO's under BCMSYS. The name of the skipped definition is printed with a message.

If an instruction or directive is illegal, this card is rejected and marked by "\*\*\*" on the printed code line.

AUTODIAGRAM recognizes four flowchart directives to modify the flowchart resulting from the instruction scan. They are to be given as comment cards with the period in column one.

1. .b\*\* - transfer list. Keep label and/or instruction counter of the next instruction card, but replace the instruction by the transfer list submitted on this card.
2. .b\*/ - add the transfer list to the next instruction.
3. .b\*+n - increment the instruction counter by the positive integer given on this card.
4. .b\*-n - skip the next n cards.

While case 4 is always honored, cases 1, 2 and 3 are used only if they appear under the proper location counter and are outside a PROC definition. The END directive which is not the end of a PROC or FUNC is used as END OF INPUT for the deck.

NOTE: There must be the same number of PROC definitions as END directives. Also, AUTODIAGRAM assumes that all DO directives are true. Hence, the technique of using a "DO" to conditionally invoke a "PROC" does not produce the same results with AUTODIAGRAM as the assembler does. It is suggested that the flowchart directive b\*-1 be used where necessary to produce the correct flowchart.

Type - PERT5

Standard PERT data cards

Type - PERT7

NASA PERT data cards

} The deck must be followed by Col. 2-6 \*\*\*\*\* as an "END" card.

Type - LOGIC

Logic is a special feature of AUTODIAGRAM. It can be used for planning a program or to give the overall documentation of a task. AUTODIAGRAM needs as information:

1. The name of the elements. They can be up to 5 alpha- numerics punched in Col. 1-6, they will be right adjusted by the scan.
2. The connection between these elements. Each name punched in the following (A6) fields, means this element is called by the first element on the card. A "+" in front of a called element declares such an element as a subroutine. The card scan stops at the first blank field or after field 12 (Col. 72).
3. Col. 2-6 \*\*\*\*\* is the "END" card for connection matrix input.
4. Comment cards.
  - Col. 1-6 - again the name of an element.
  - Col. 8-42 - the description for this element, which will appear as comment inside the box.
5. Col. 2-6 \*\*\*\* is the "END" card for the description cards.

NOTE: There must be at least one comment card (which may be a blank card).

SUMMARY AND CONCLUSIONS

AUTODIAGRAM is a useful tool to analyze and to plan system or application programs. It can be used at any stage in the development of the program. It is easy to use and the results are easy to read. On request, loops can be displayed in a hierarchical order, or the flowchart can be split to limit the results to a manageable size.

The picture of the resulting flowchart is independent of the source input type, so that it is possible to compare the logical structure of FORTRAN decks and UNIVAC assembly language programs, if they are written for the same tasks. If the input feature LOGIC is used for planning a program, the flowchart of the resulting program must then logically agree with the flowchart given by LOGIC. By comparing the flowcharts of different versions of the same routine, "insignificant" alterations which have changed the logical structure of the deck can be found.

The use of AUTODIAGRAM saves debugging time and effort.

1032-GIH-mjg

*G. I. Hain*  
G. I. Hain

**BELLCOMM, INC.**

Subject: AUTODIAGRAM Version for  
UNIVAC 1108 - Case 803

From: G. I. Hain

Distribution List

NASA Headquarters

Messrs. K. R. Joseph - NASA/TN (10)  
T. A. Keegan - NASA/MA

Bellcomm

Messrs. A. P. Boysen, Jr.  
K. R. Carpenter  
J. O. Cappellari, Jr.  
D. R. Hagner  
W. G. Heffron  
H. A. Helm  
W. C. Hittinger  
B. T. Howard  
C. M. Klingman  
J. Z. Menard  
V. S. Mummert  
I. D. Nehama  
J. M. Nervik  
I. M. Ross  
R. V. Sperry  
R. L. Wagner

Computer Bulletin Distribution List (Bellcomm Only)  
Central File  
Department 1023  
Library