221
10-7

AEC RESEARCH AND DEVELOPMENT REPORT

DWR-1580

Y-1735

Mathematics and
Computers

MASTER

COMPUTER ROUTINES FOR EDITING
NUMERICAL CONTROL
META-LANGUAGE

C. W. Wilson

UNION CARBIDE CORPORATION
NUCLEAR DIVISION
OAK RIDGE Y-12 PLANT

operated for the ATOMIC ENERGY COMMISSION under U. S. GOVERNMENT Contract W-7405 eng 26

UNION
CARBIDE

OAK RIDGE Y-12 PLANT
P. O. Box Y
OAK RIDGE, TENNESSEE 37830

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Printed in the United States of America. Available from Clearinghouse for Federal
Scientific and Technical Information, National Bureau of Standards,
U.S. Department of Commerce, Springfield, Virginia 22151
Price: Printed Copy \$3.00; Microfiche \$0.65

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Date Issued: September 28, 1970

Document Y-1735

TID-4500

UNION CARBIDE CORPORATION
Nuclear Division

OAK RIDGE Y-12 PLANT

Operated under Contract W-7405-eng-26
With the US Atomic Energy Commission

COMPUTER ROUTINES FOR EDITING NUMERICAL
CONTROL META-LANGUAGE

C. W. Wilson

LEGAL NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Oak Ridge, Tennessee

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

Document Y-1735

TID-4500

DISTRIBUTION

Atomic Energy Commission

Keller, C. A.
Zachry, D. S., Jr

Bendix Industrial Controls

Struckel, Marlene

Burroughs Corporation

Bellows, Harry

Cincinnati Milacron

Thiel, Mick

Control Data Corporation

Sanchez, Louis

Douglas Aircraft Company

Lowe, Gene

General Electric Company

Crowley, R. E.

IBM Development Center

Pinter, H. R.

IIT Research Institute

Rogge, Susanne

Kearney and Trecker Corporation

Grenlie, A. H.

Lockheed-Georgia Company

Wheeler, F. D.

Oak Ridge Gaseous Diffusion Plant

Jordan, R. G.
Wilcox, W. J., Jr

Oak Ridge Y-12 Plant

Beeson, H. C.
Bell, B. B.
Bernander, N. K.
Briscoe, O. W.
Burkhart, L. E.
Denny, A. (2)
Ellingson, R. D.
Evans, G. W.
Foulk, D. L.
Haeusler, K. R.
Hemphill, L. F.
Jackson, V. C.
Kahl, K. G.
Keith, Alvin
Kite, H. T.
McLendon, J. D.
Mitchel, G. W.
Oliphant, G. W.
Phillips, L. R.
Smith, R. D.
Stoner, H. H.
Trotter, T. C.
Waters, J. L.
Wesley, R. L.

Wilson, C. W. (25)
Yaggi, W. J.
Y-12 Central Files (5)
Y-12 Central Files (route)
Y-12 Central Files (master copy)
Y-12 Central Files (Y-12RC)

Paducah Gaseous Diffusion Plant

Winkel, R. A.

Sperry Rand Corporation

Mangold, W. E.

Tinker Air Force Base

Wood, W. J.

TRW System, Incorporated

Skeen, J. R.

University Computing Company

Fowler, A. H.

In addition, this report is distributed in accordance with the category Mathematics and Computers, as given in the "USAEC Standard Distribution Lists for Unclassified Scientific and Technical Reports", TID-4500.

ABSTRACT

A meta-language technique permits rigorous and complete definition of a computer language. The technique is applicable to almost any language.

CONTENTS

SUMMARY	7
INTRODUCTION	8
THE META-LANGUAGE	9
Technique.	9
Input Data	9
Computer Program	11
APPENDIX A	15
Input Card Layouts	15
APPENDIX B	20
Sample Input	20
APPENDIX C	21
Sample Output.	21
APPENDIX D	29
Computer Load Map	29
APPENDIX E	31
Program Flow Charts	31
APPENDIX F	38
Program Lists	38

SUMMARY

A meta-language technique permits rigorous and complete definition of a computer language. Although developed to assist Subcommittee ANSI/X3.4.7 in its preparation of a proposed standard for the APT (numerical control) language, the computer program has been generalized sufficiently to permit languages other than APT to be included.

INTRODUCTION

Subcommittee X3.4.7 of the American National Standards Institute (ANSI) is charged with the responsibility of creating a proposed draft of a proposed standard for the APT numerical control language. The subcommittee is describing APT as a meta-language of positive numbers where each number represents an element in the language. This approach has the advantage of permitting use of the extensive manipulative ability of the digital computer. The system of computer programs described in this report is used as a tool by the subcommittee in its task of standardization.

The International Standard Organization Subcommittee TC97/SC5/WG1, which is working on an international standard of a numerical control language, is using this program to check the internal consistency of candidate languages and to compare the syntax of the various candidates.

THE META-LANGUAGE

TECHNIQUE

The syntax for the language is defined using a meta-language of positive integers. Each meta-integer of the meta-language represents a syntactic element of the language. The meta-integer assigned to a syntactic element of the language must be unique, but its value is arbitrary. Once a syntactic element has been associated with a meta-integer, all references to that element must use that meta-integer. Each meta-integer is a syntax rule containing one or more subrules. Each subrule specifies a particular string of one or more meta-integers (rules). Each string (subrule) is one of the syntactic definitions of the original meta-integer (rule). Each meta-integer (rule) is given a descriptive name. The semantics contain all pertinent comments required for the interpretation of the associated meta-integer (rule).

It was originally decided to assign the values of the integers in a logical order, where higher integers implied a higher level of syntax. However, it was soon proved that it was more practical and expedient to assign blocks of integers to different individuals who were simultaneously working on different sections of the language in widely scattered locations. Reassignment of the integers in a more logical pattern will be trivial at some later time when the logical pattern becomes more apparent.

The meta-integers are considered to be divided into several sections, the principal division being between the structural and basic syntactic elements. The meta-integers from 1 to 2000 represent structural syntax; meta-integers from 2000 to 3000 represent the basic syntax of the language.

The basic syntax section is divided into three subsections, namely: basic characters, punctuation, and vocabulary words. The vocabulary is divided further into two subsections of two parts each. The processor words relate to geometry. Their interpretation is independent of the type of numerically controlled machine that is the target of a program. The postprocessor words relate to and their interpretation depends upon the numerically controlled machine that is the target of a program. Both processor and postprocessor words are classified as either major or minor. A major word always has the same meaning; a minor word has an interpretation that is conditional upon the major word with which it appears.

INPUT DATA

There are four types of input data (see Table 1).

The rule numbers are selected from the set of positive integers between 1 and 3000 and must be right adjusted in their fields. These 3000 rules are divided as given in Table 2.

Table 1
TYPES OF INPUT DATA

Item	Card Type	Figure (1)
1. Setup		
a. Operation	4	A-1
b. Subject Language Flags	5	A-2
c. Subject Language Name	6	A-3
2. System Flags	7	A-2
3. Variable Name and Semantics	8	A-4
4. Syntax	9	A-5

(1) See Appendix A.

Table 2
DIVISION OF RULE NUMBERS

Rule Number	Syntax
1-2000	Structural Elements
2001-2128	Basic Characters
2129-2400	Processor Major Vocabulary
2401-2599	Processor Minor Vocabulary
2600-2788	Postprocessor Major Vocabulary
2789-3000	Postprocessor Minor Vocabulary

Sample input cards are shown in Appendix B; sample output forms are shown in Appendix C.

The subrule numbers are selected from the set of integers: -2 and 1 through 99, and must be right adjusted in their field. Only Subrule 1 may accompany Subrule -2 and indicates an "is defined as" "but does not include" condition, respectively. The subrule numbers must be unique, but not necessarily consecutive, within a given rule. The continuation field contains blank or zero, 1 through 9, and A through Z. They should be in the specified order and consecutive within each rule-subrule combination. The card type should be punched in each card. The date field is optional and is replaced on the master data file with the date of the computer run in which that card was added or altered. However, it will be listed in the update card list and hence provides a useful tool to the user to more closely identify when the update card was created. The update flag field is used to indicate that a certain card image on the master data file is to be deleted, "1", or that the end of the update deck has been reached, "9". Any other value assumes that a change or addition is desired.

The "operation" setup card specifies the options desired for updating, listing, and punching the master data file. The hierarchy of operations is: (1) create a new master file from cards, (2) update, and (3) produce lists. The first two items may be performed once, at most, on a computer run. The list-producing level may be repeated. The digit "1" in the appropriate column causes that operation to be performed. The "list variable name" field (Column 8) may contain the digit "2" to produce a list of the complete semantics deck. The "numerical control meta-language report" field may contain the digit "2" to produce a list that is 132 columns wide. If the "compare 2 subject languages" flag (Column 16) is set, the only operable flags are "tape to card", "card to tape", and "update" (Columns 13, 14, 15). Also, a second "subject language flags" setup card and "subject language name" setup card

must be provided. If "tape to card" is selected, the lower and upper rule numbers of a range can be specified in Columns 17 - 20 and 21 - 24, respectively. If either is blank, all rules are punched. If a range is selected, the desired card types are indicated by the digit "1" in the appropriate column (see Appendix A, Figure A-1).

The "subject language flags" setup card specifies the systems, subsets, and modules to be considered in the subject language. The digit "1" in the proper column turns on the flags for all syntax items in that system. Any number of flags may be turned on. The flags are all turned off before each set of setup cards is read.

The "subject language name" setup card specifies the name of the subject language composed of the combination of systems, subsets, and modules specified in the "subject language flags" setup card. It may contain any standard alphanumeric and special characters. However, the card must be present even if it is blank.

The "system flags" card specifies the systems, subsets, and modules in which the rule-subrule combination is contained. All rule-subrule combinations are considered to be members of the reference system. The digit "1" in the proper column indicates that the rule-subrule combination is a member of that system, subset, or module; the digit "0" means it is not. When updating, only the changed columns need be punched since blank columns are ignored for Card Type 7.

The "variable name and semantics" cards contain the name of the meta-linguistic variable in the first card for each rule number and any optional semantic comments about that rule in the following cards. The first card must be designated as Subrule "0" and the subrule field of each of the following semantics cards should contain the subrule to which that comment applies. Continuation cards are accepted, so general comments applying to the entire rule could be contained in continuation cards of Subrule "0". Each subrule may contain the full quota of continuation cards, but the semantics should be terse. The information should be left adjusted to Column 9 in all cards.

The "syntax" card specifies the elements of the language which together define that rule-subrule combination. These elements are rule numbers which are also members of the subject language. The elements must be right adjusted in their respective fields.

COMPUTER PROGRAM

The programs are written in Fortran and Cobol language for the IBM System 360. The IBM input/output and function library are assumed to be available (see Appendix D). The loaded system requires about 325,000 bytes of core storage. Two non-standard tapes are required. Logical Tape 1 is considered to be the current master data file; Logical Tape 2 will contain the updated master data file if there is updating on a given computer run. The tapes are written in hollerith format as card

images of the input data cards described in the input data section. Each deck of card types is terminated by a record of Rule Number =9999, Subrule =99, and Continuation =9. On a Model 91, the program has rarely run as long as four minutes. Program flow charts and lists are shown in Appendixes E and F, respectively. A sample load map is shown in Appendix D.

The program MAIN consists of 264 Fortran statements and reads Card Types 4, 5, and 6 from the input stream. The routines for "card to tape", "update", and "tape to card" are performed as required. The program then sets up variable names and syntax tables in core for those elements contained in the subject language and lists Card Types 7, 8, and 9 if required. The system eliminates references to rules, containing only one subrule whose sole element is the meta-integer representing null. These cases are created most frequently when subsets or modules of a fully defined and internally consistent language are specified as the subject language. The cross reference, comparison, abbreviated list, and full list are then produced as required by Card Type 4. The entire process may then be repeated, with the exception of the card-to-tape and update portions which may be used only once per computer run.

APTCTT consists of 40 Fortran statements and reads the complete data deck of Card Types 7, 8, and 9 from the input data stream and writes it on Logical Tape 1. The cards are listed as read, and the number of cards read is printed when the end update flag is reached.

APTTTC consists of 39 Fortran statements and punches the complete master file into cards. If updating has occurred, Logical Tape 2 will be the master data file source. Otherwise, Logical Tape 1 will be used. The cards are listed as punched, and the number of cards punched will be written when an end-of-file point is reached.

BLOCK consists of 15 Fortran statements and initializes those variables in named common which require the same value throughout the program.

COMSYS consists of 47 Fortran statements and compares the syntax of the two subject languages defined in the two sets of Card Types 5 and 6. The report lists every rule-subrule combination in which the two systems differ. The inclusion of a particular rule-subrule combination in a system is indicated by an "X" in the appropriate column. The variable name is listed for convenience.

LONG consists of 96 Fortran statements and edits the meta-language description of the subject language into a more readable form. Each rule is identified by its variable name. Each element in the string of elements defining a subrule is identified by its meta-integer and variable name in a form similar to the Backus Normal form. It has the option of printing a page that is either 8 1/2 or 14 inches wide. This option is set in the proper field of Card Type 4 and is described in the input data

section. Because of the verbosity of this report, it is frequently referred to as the "long form".

MODULE consists of 60 Fortran statements and lists the subsets or the modular feature of which each rule-subrule combination is a member.

SHORT consists of 143 Fortran statements and serves the same function as LONG. However, it is an abbreviated form since those meta-integers in a subrule string of elements which are basic symbols to the language (rule number ≥ 2000) are replaced by the character, or characters, specified by the syntax of that basic symbol in its first subrule in the subject language. That is, meta-integers for the basic characters and the vocabulary will be replaced by the basic character or vocabulary word. In addition, the date of the last change is printed for each subrule on the right edge of the report. Because this report tends to be more abbreviated, it is frequently referred to as the "short form".

UPDATE consists of 112 Fortran statements and merges Card Types 7, 8, and 9 from the input stream into the master data file from Logical Tape 1, deleting, replacing, changing, and adding as required, and places the resulting master data file on Logical Tape 2. The cards from the input stream are listed as read. Any card in the input stream not in strictly increasing order of card type-rule-subrule-continuation will be ignored for updating and the user will be so informed in the list. This routine may be used only once on each computer run.

XREFER consists of 69 Fortran statements. For each syntax rule, it lists every rule-subrule combination in which that rule appears. This program can be a very useful tool for updating. As the number of rules and subrules in the reference system increases, it becomes increasingly mandatory that updates be verified using such a report as this to assure consistency and protection in the system.

APTSTR consists of 90 Cobol statements and has the entry points INSERT (which inserts characters in a string of characters), INTBCD (which converts Fortran integers to hollerith), COMPAR (which compares a specified number of bytes for equivalence), and FETCH (which extracts a character from a string of characters). The principal reasons for using Cobol for these routines are its integer-to-hollerith capability and the ease of character manipulation.

IDAY is a local assembler language program which extracts the current date from a core location. The only argument of this routine will contain the date in hollerith form.

The variables used by more than one of the routines are stored in named common areas, namely; COM, OTHER, DATES, and CONTNO. Any initialization of the common variables is performed in BLOCK DATA.

The principal variables are summarized in Table 3.

Table 3
PRINCIPAL VARIABLES

Variable	Use
ASCII	Contains the graphic representation of the basic characters.
CONTIN	Contains the continuation characters in their accepted order.
POINTR	Contains subscripts and quantities for each rule for the arrays SEMANT, SYNTAX, and SYSTEM.
SEMANT	Contains the variable names as a continuous string. The subscript for the first character in SEMANT for Rule 1 is stored in POINTR (1, 4) and the number of characters is stored in POINTR (1, 5).
SEMDAT	Contains the last update date for each rule variable name.
SYNDAT	Contains the last update date of the syntax for each subrule of each rule. The pointer to the proper location in SYNDAT is stored in SYNTAX.
SYNTAX	Contains the syntax as a string of integers. The subscript for the first entry in SYNTAX for Rule 1 is stored in POINTR (1, 1). The order of integers for Rule 1 as stored in SYNTAX is: number of subrules, first subrule number, SYNDAT subscript, number of elements in first subrule, string of elements for first subrule, . . . , last subrule number, SYNDAT subscript, number of elements in last subrule, string of elements for the last subrule.
SYSTEM	Contains the subrule numbers contained in the subject language with their associated subset and modular feature numbers. The subscript for the first entry in SYSTEM for Rule 1 is stored in POINTR (1, 2) and the number of subrules stored in SYSTEM for Rule 1 is stored in POINTR (1, 3).

APPENDIX A

INPUT CARD LAYOUTS

1.			41.
2.			42.
3.	Not Used		43.
4.			44.
5.			45.
6.	List Modular Features		46.
7.	List System Flags		47.
8.	List Variable Name		48.
9.	List Syntax		49.
10.	Cross Reference Report		50.
11.	N/C Meta-Language (abbrev.)		51.
12.	N/C Meta-Language Report		52.
13.	Punch Master Data File		53.
14.	Make Master File from Cards		54.
15.	Update Master Data File		55.
16.	Compare 2 Subject Languages		56. Not Used
17.			57.
18.	Lower Rule	Rule	58.
19.		Punch	59.
20.		Selector	60.
21.	Upper Rule		61.
22.			62.
23.			63.
24.			64.
25. 7	Card		65.
26. 8	Type		66.
27. 9	Selector		67.
28.			68.
29.			69.
30.	Not Used		70.
31.			71.
32.			72. 4 (Card Type)
33.			73.
34.			74.
35.			75. Date
36.			76.
37.			77.
38.			78.
39.			79.
40.			80.

Figure A-1. "OPERATION" SETUP CARD. (Card Type 4; Format 6X, 1011, 55X, 11, A8)

(CT7)		(CT5)	
1.		Not Used	41. (3.5) DNTCUT-CUT
2. Rule Number			42. (3.6) Region Capability
3.			43. (3.7) Geom. Definition Manipulation
4.			44. (3.8) Tabcyl Definition and Contouring
5. Not Used			45. (3.9) Additional Matrix Definitions
6. Subrule			46. (3.10) Advanced Matrix-Geometry
7. Numbers			47. (3.11) Advanced MACRO
8. Update Flag			48. (4.1) Generalized Cutter Capability
9. Reference			49. (4.2) Additional Geometry
10. Current APT			50. (4.3) Additional Vector Definitions
11. Proposed APT (X3.4.7)			51. (4.4) Multiple Direction Control Statements
12.			52. (4.5) Additional Cutter-Part Control
13. APT Subset 1: Minimum			53. (4.6) Ruled Surface Capability
14. APT Subset 2: Point-to-Point			54. (4.7) Individual Surface Tolerances
15. APT Subset 3: Minimum Contouring			55.
16. APT Subset 4: 3-Axis Contouring			56.
17. APT Subset 5: 5-Axis Contouring			57.
18.			58.
19.			59.
20.			60.
21. (1.1) Postprocessing Control			61.
22. (1.2) COPY-No Matrix Definitions			62.
23. (2.1) Oper. Using Matrix Definitions			63.
24. (2.2) MACRO			64.
25. (2.3) Nested Definition			65.
26. (2.4) Sp. Processor Output Verification			66.
27. (2.5) Sp. Input-Output Capability			67.
28. (2.6) Point-Pattern Capabilities			68.
29. (2.7) Adv. Point-Pattern Capabilities			69.
30. (2.8) Multiple Postprocessing Control			70.
31. (2.9) Computing			71.
32. (2.10) Looping			72. 5 or 7 (Card Type)
33. (2.11) Subscripted Identifiers			73.
34. (2.12) Additional Geom. Definitions			74.
35. (2.13) Additional Copy Capability			75. Date
36. (2.14) Synonyms			76.
37. (3.1) Conic Section Def. and Contouring			77.
38. (3.2) Loft Conic Definitions			78.
39. (3.3) Surface Thickness Capability			79.
40. (3.4) Cond. Check Surface Transfer			80.

Figure A-2. "SUBJECT LANGUAGE FLAGS" SETUP CARD AND "SYSTEM FLAGS" CARD.
(Card Types 5 and 7; Format 14, 1X, 12, 6511, A8)

1.	41.
2.	42.
3.	43.
4.	44.
5.	45.
6. Subject Language	46.
7.	47.
8.	48.
9.	49.
10.	50.
11.	51.
12.	52.
13.	53.
14.	54.
15.	55.
16.	56.
17.	57.
18.	58.
19.	59.
20.	60.
21.	61.
22.	62.
23.	63.
24.	64.
25.	65.
26.	66.
27.	67.
28.	68.
29.	69.
30.	70.
31.	71.
32.	72. 6 (Card Type)
33.	73.
34.	74.
35.	75. Date
36.	76.
37.	77.
38.	78.
39.	79.
40.	80.

Figure A-3. "SUBJECT LANGUAGE NAME" SETUP CARD. (Card Type 6; Format 8A8, A7, 11, A0)

1.	41.
2.	42.
3. Rule Number	43.
4.	44.
5. Continuation	45.
6. Subrule Number	46.
7.	47.
8. Update Flag	48.
9.	49.
10.	50.
11.	51.
12.	52.
13. Variable Name	53.
14.	54.
15. or	55.
16.	56.
17. Semantics	57.
18.	58.
19.	59.
20.	60.
21.	61.
22.	62.
23.	63.
24.	64.
25.	65.
26.	66.
27.	67.
28.	68.
29.	69.
30.	70.
31.	71.
32.	72. 8 (Card Type)
33.	73.
34.	74.
35.	75. Date
36.	76.
37.	77.
38.	78.
39.	79.
40.	80.

Figure A-4. "VARIABLE NAME AND SEMANTICS" CARD. (Card Type 8; Format 14; A1, 12, 11, 63A1, 11, A8)

1.		41.
2.		42.
3. Rule Number		43. Field 9
4.		44.
5. Continuation		45.
6. Subrule Number		46.
7.		47. Field 10
8. Update Flag		48.
9.		49.
10.	Syntax	50.
11. Field 1		51. Field 11
12.		52.
13.		53.
14.		54.
15. Field 2		55. Field 12
16.		56.
17.		57.
18.		58.
19. Field 3		59. Field 13
20.		60.
21.		61.
22.		62.
23. Field 4		63. Field 14
24.		64.
25.		65.
26.		66.
27. Field 5		67. Field 15
28.		68.
29.		69.
30.		70. Not Used
31. Field 6		71.
32.		72. 9 (Card Type)
33.		73.
34.		74.
35. Field 7		75. Date
36.		76.
37.		77.
38.		78.
39. Field 8		79.
40.		80.

Figure A-5. "SYNTAX" CARD. (Card Type 9; Format 14, 11, 12, 11, 1514, 3X, 11, A8)

APPENDIX B

SAMPLE INPUT

```

//CWWA JOB 11656,'C W WILSON',MSGLEVEL=1,REGION=470K,CLASS=E
//APTSYN EXEC F0RTHLG,REGION=470K
//LINK.SYSIN DD *
INSERT HEX DECKS HERE
//G0.FT01F001 DD UNIT=TAPE9,LABEL=(,BLP),VOLUME=SER=1,DSNAME=0LDSYN,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600),DISP=(OLD,KEEP)
//G0.FT02F001 DD UNIT=TAPE9,LABEL=(,BLP),VOLUME=SER=2,DSNAME=NEWSYN,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600),DISP=(OLD,KEEP)
//G0.FT07F001 DD SYSOUT=B
//G0.FT06F001 DD SYSOUT=A
//G0.FT05F001 DD *
      121112 1
1
ANSI/X3.4.7 REFERENCE LANGUAGE
15 1 01111
68 1 101
68 2 101
68 31
699 1 111
2207 1 101
2207 5 101
699 0 REFER TO SEMANTICS
6991 0 THIS META-LINGUISTIC VARIABLE IS USED IN CERTAIN CASES
6992 0 WHERE RIGOROUS USE OF META-INTEGERS WOULD PRODUCE SYNTAX OF
6993 0 EXCESSIVE LENGTH AND THE SYNTAX CAN BE ADEQUATELY DEFINED IN
6994 0 THE SEMANTICS
2207 0 SYM FOR ARRAY
476 2 2140 4772144 152148 312144 31 1412141
699 1 699
2207 1 20652082208220652089
2207 5 15
9
1
1
CURRENT APT
1
PROPOSED APT
/*

```

4
5
6
7
7
7
7
7
7
7
8
8
8
8
8
8
9
9
9
9
4
5
6
5
6

NUMERICAL CONTROL META-LANGUAGE SUBSETS AND MODULES 05-19-70 PAGE- 2

IN SYSTEM-00001000
SYSTEM NAME-PROPOSED APT STANDARD : MINIMUM SUBSET

RULE	SUBRULE	SUBSETS	MODULAR-FEATURE NO. MIN. SUBSET
------	---------	---------	------------------------------------

128	1	1	2	3	4	5
132	1	1	2	3	4	5
133	1	1	2	3	4	5
134	1	1	2	3	4	5
141	2	1	2	3	4	5
201	1	1	2	3	4	5
	2	1	2	3	4	5
	3	1	2	3	4	5
203	1	1	2	3	4	5
	2	1	2	3	4	5
204	1	1	2	3	4	5
	3	1	2	3	4	5
205	1	1	2	3	4	5
	2	1	2	3	4	5
206	1	1	2	3	4	5
	2	1	2	3	4	5
207	1	1	2	3	4	5
	2	1	2	3	4	5
380	1	1	2	3	4	5
	2	1	2	3	4	5
	3	1	2	3	4	5
	4	1	2	3	4	5
382	1	1	2	3	4	5
453	18	1				
500	1	1	2	3	4	5
502	1	1	2	3	4	5
	3	1	2	3	4	5
	5	1	2	3	4	5
505	1	1	2	3	4	5
601	1	1	2	3	4	5
602	1	1	2	3	4	5
	4	1	2	3	4	5
603	1	1	2	3	4	5
	2	1	2	3	4	5
607	2	1	2	3	4	5
	4	1	2	3	4	5
	10	1	2	3	4	5
	14	1	2	3	4	5
610	1	1	2	3	4	5
617	1	1	2	3	4	5
	2	1	2	3	4	5
623	1	1	2	3	4	5
	2	1	2	3	4	5
630	1	1	2	3	4	5
	2	1	2	3	4	5
631	3	1	2	3	4	5
800	1	1	2	3	4	5
802	1	1	2	3	4	5
	3	1	2	3	4	5
	4	1	2	3	4	5
	10	1	2	3	4	5
	13	1	2	3	4	5
	14	1	2	3	4	5

MODULAR-FEATURE
NO. . MIN. SUBSET

2.1 2

UPDATED

< 19 UNSIGNED NUMBER>	07-11-69
IS(1) <12 STRING OF DIGITS>	03-22-68
OR(2) <12 STRING OF DIGITS>.	10-31-69
OR(3) .<12 STRING OF DIGITS>	10-31-69
OR(4) <12 STRING OF DIGITS>.<12 STRING OF DIGITS>	10-31-69
< 31 SCALAR>	07-11-69
IS(1) <37 PRIMARY>	03-22-68
OR(2) <17 ADDING OPERATOR><37 PRIMARY>	03-22-68
< 37 PRIMARY>	07-11-69
IS(1) <19 UNSIGNED NUMBER>	03-22-68
< 50 SUBSET OF KEYWORDS>	07-11-69
IS(1) FINI	03-22-68
< 120 CLOCK DIRECTION SEL>	03-20-70
IS(1) CLW	11-19-68
OR(2) CCLW	11-19-68
< 126 KEY WORD>	02-03-70
IS(1) <699 >	03-20-70
< 128 NULL>	02-03-70
IS(1) <699 >	03-20-70
< 132 TWO SCALARS>	03-22-68
IS(1) <31 SCALAR>,<31 SCALAR>	03-22-68
< 133 THREE SCALARS>	03-22-68
IS(1) <132 TWO SCALARS>,<31 SCALAR>	03-22-68
< 134 FOUR SCALARS>	03-22-68
IS(1) <133 THREE SCALARS>,<31 SCALAR>	03-22-68
< 201 STATEMENT TERMINATION>	10-31-69
IS(1) <205 COMMENT (OPT)><203 END OF LINE>	02-03-70
OR(2) <205 COMMENT (OPT)>;	02-03-70
OR(3) <209 >;<206 LITERAL CHAR STRING><203 END OF LINE>	02-03-70
< 203 END OF LINE>	03-22-68
IS(1) <2110 SIGNAL FOR END OF LINE>	10-31-69
OR(2) <2109 SIGNAL FOR BEGIN LINE ID>	02-03-70
<204 LINE IDENTIFICATION (OPT)>	
<2110 SIGNAL FOR END OF LINE>	
< 204 LINE IDENTIFICATION (OPT)>	07-11-69
IS(1) <128 NULL>	03-22-68
OR(3) <206 LITERAL CHAR STRING>	03-22-68

NUMERICAL CONTRCL META-LANGUAGE 05-19-70 PAGE- 2

IN SYSTEMS-00001000
SYSTEM NAME-PROPOSED APT STANDARD : MINIMUM SUBSET

```

< 19 UNSIGNED NUMBER>
  IS( 1) <12 STRING OF DIGITS>
  OR( 2) <17 STRING OF DIGITS><2146 SYM FOR DECIMAL POINT>
  OR( 3) <2146 SYM FOR DECIMAL POINT><12 STRING OF DIGITS>
  OR( 4) <12 STRING OF DIGITS><2146 SYM FOR DECIMAL POINT>
    <12 STRING OF DIGITS>

< 31 SCALAR>
  IS( 1) <37 PRIMARY>
  OR( 2) <17 ADDING OPERATOR><37 PRIMARY>

< 37 PRIMARY>
  IS( 1) <19 UNSIGNED NUMBER>

< 50 SUBJECT OF KEYWORD3>
  IS( 1) <2244 SYM FOR END OF PART PROGRAM>

< 120 CLOCK DIRECTION SEL>
  IS( 1) <2413 SYM FOR CLOCKWISE>
  OR( 2) <2411 SYM FOR COUNTER CLOCKWISE>

< 126 KEY WORD>
  IS( 1) <699 >

< 128 NULL>
  IS( 1) <699 >

< 132 TWO SCALARS>
  IS( 1) <31 SCALAR><2144 SYM FOR WORD SEPARATOR><31 SCALAR>

< 133 THREE SCALARS>
  IS( 1) <132 TWO SCALARS><2144 SYM FOR WORD SEPARATOR><31 SCALAR>

< 134 FOUR SCALARS>
  IS( 1) <133 THREE SCALARS><2144 SYM FOR WORD SEPARATOR><31 SCALAR>

< 201 STATEMENT TERMINATION>
  IS( 1) <205 COMMENT (OPT)><203 END OF LINE>
  OR( 2) <205 COMMENT (OPT)><2159 SYM FOR END OF STAT>
  OR( 3) <209 ><2159 SYM FOR END OF STAT><206 LITERAL CHAR STRING>
    <203 END OF LINE>

< 203 END OF LINE>
  IS( 1) <2110 SIGNAL FOR END OF LINE>
  OR( 2) <2109 SIGNAL FOR BEGIN LINE ID><204 LINE IDENTIFICATION (OPT)>
    <2110 SIGNAL FOR END OF LINE>

< 204 LINE IDENTIFICATION (OPT)>
  IS( 1) <128 NULL>
  OR( 3) <206 LITERAL CHAR STRING>

```

NUMERICAL CONTROL META-LANGUAGE COMPARISON 05-19-70 PAGE- 2

SYSTEMS 1 -00001000

SYSTEM NAME-PROPOSED APT STANDARD : MINIMUM SUBSET

SYSTEMS 2 -00000100

SYSTEM NAME-PROPOSED APT STANDARD : PT TO PT SUBSET

RULE	SUBRULE	1	2	(X MEANS IN SYSTEM)
454	1	X	LINE PAR.STRING	
	2	X		
	3	X		
	4	X		
	5	X		
	6	X		
	15	X		
	19	X		
455	1	X	CIRCLE PAR.STRING	
	2	X		
	3	X		
	4	X		
	5	X		
	8	X		
	9	X		
	13	X		
	14	X		
	18	X		
	19	X		
460	1	X	PLANE PAR.STRING	
607	1	X	ORDINARY COMMAND	
	5	X		
800	2	X	POSTPROCESSOR COMMAND	
803	1	X	STRAIGHT LINE MILLING AND DRILLING POSTPROCESSOR COMMAND	
	2	X		
	3	X		
805	7	X	SINGLE HEAD LATHE POSTPROCESSOR COMMAND	
808	2	X	OPTIONAL POSTPROCESSOR STATEMENTS CLASS SPEEDS/FEEDS	
909	1	X	GNE OF- ,CLW ,CCLW	
	2	X		
937	1	X	FEDRAT PARAMETER LIST	
	2	X		
993	1	X	SPINDL PARAMETER LIST	
	2	X		
2212	1	X	SYM FOR CIRCLE DEF FCN	
	5	X		
2286	1	X	SYM FOR LINE DEF FCN	
	5	X		
2330	1	X	SYM FOR PLANE DEF FCN	
	5	X		
2334	1	X	SYM FOR POINT DEF FCN	
	5	X		
2400	1	X	SYM FOR Z COORDINATE SURFACE	
	5	X		
2402	1	X	SYM FOR ATANGL	
	5	X		
2403	1	X	SYM FOR CANON	
	5	X		
2404	1	X	SYM FOR CENTER	
	5	X		

COMPUTER LOAD MAP

APPENDIX D

CONTROL SECTION

NAME	ORIGIN	LENGTH
MAIN	00	1390
COM	1080	2AE64
CONTNO	2C6E3	25
APICIT	2C710	394
APITTC	2CAA8	364
COMSYS	2CE10	44C
LONG	2D260	820
MODULE	2DA80	680
SHORT	2E100	8FE
UPDATE	2EDC0	834
XREFER	2F833	6A0
APTSTR	2FED8	992
IHCPIXPI*	30870	A8
IHCFCUMH*	30918	F3C
IDAY *	31853	108
IHCFCVTH*	31960	107C
ARITH= *	329E0	473
IHCFIUSH*	32E58	EB6
DUMPDUMP*	33010	6A8
DUMDUM *	34388	1666
GDUMP *	35A20	450
PUTOUTG *	35E70	780
RDLQADM *	36620	98F
IHCUTATBL*	36FE0	398
SETDUMP *	37378	668
IHCSEXIT*	379E0	1C
ICOMPARE*	37A00	6C
INTOBCD *	37A70	10E
IWBGZZZZ*	37880	80
LOC *	37C00	A
REPLACE *	37C10	50
OTHER	37C60	90
DATES	37CF0	157C0
WHICH	40480	14
DUMPCODS	404C8	30
CSECTS\$\$	404F8	12C4

ENTRY

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
INSERT	30238	INTHOL	30458	COMPAR	3059C	FETCH	30644
FIXPI=	30870						
IBCOM=	30918	IBCOM	30918	FDIOWS=	3090C	INTSW	3159D
DUMPFLAG	315A0						
ADCON=	31960	FCVZJ	314AC	FCVAJ	31852	FCVLO	318E2
FCVIO	31F18	FCVEJ	3240A	FCVCJ	3260C	INTSSW	329C1
SSSTIMES	328F6	ORNLTNTH	32C44				
FIOCS=	32E58	SETFID	33C98	SETSYSQU	33CA8		
SETFAU	3740A	SETFAULT	37446				
EXIT	379E0						
ICOMPA	37A00						
INTOBC	37A70	INTBOLL	37A84	INTBOD	37A84		

ENTRY ADDRESS 00
TOTAL LENGTH 4E7C0

***NAMEX DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX E

PROGRAM FLOW CHARTS

Figure E-1. PROGRAM "MAIN".

Figure E-2. PROGRAM "APTCTT".

Figure E-3. PROGRAM "APTTTC".

Figure E-4. PROGRAM "COMSYS".

Figure E-5. PROGRAM "LONG".

Figure E-6 PROGRAM "SHORT".

Figure E-7. PROGRAM "UPDATE".

Figure E-8. PROGRAM "XREFER".

APPENDIX F

PROGRAM LISTS

```

      IMPLICIT INTEGER*4 (A-Z)
      COMMON /COM/ ASCII(128),DATE,SEMANT(25000),FLAGS(20),SYNTAX(20000)
1    1,POINTR(3000,5),SYSTEM(5000),INTAPE,WORDIN(62),WORDOT(62)
      COMMON /OTHER/ SYSNAM(9,2)
      COMMON /DATES/ SEMDAT(3000),SYNDAT(8000)
      COMMON /WHICH/ IPUNW(5)
      REAL*8 SEMDAT,SYNDAT
      LOGICAL*1 SEMANT,ASCII
      REAL*8 DATE,SYSNAM
      INTEGER*2 SYSTEM,SUBRUL
      REAL*8 BLANK/' ',OD,AR(12),SEMNAM(2)/'VARIABLESEMANTIC'/
      DIMENSION IT(15),WORD(62)
      LOGICAL*1 BLNK1/' ',DESC(63)
      CALL IDAY (DATE)
      INTAPE=1
10   READ (5,1000,END=300)(FLAGS(I),I=6,16),IPUNW
1000 FORMAT(5X,11I1,2I4,3I1)
      READ (5,1001) WORDIN
1001 FORMAT(8X,62I1)
      READ (5,1016) (SYSNAM(I,1),I=1,9)
1016 FORMAT(8A8,A7)
      IF( FLAGS(16) .EQ. 0 ) GO TO 11
      READ (5,1001) WORDOT
      READ (5,1016) (SYSNAM(I,2),I=1,9)
11   SYSIN1=0
      SYSOT1=0
      DO 12 I=1,31
        I1=2**(I-1)
        SYSIN1=SYSIN1+WORDIN(I)*I1
12   IF( FLAGS(16) .EQ. 1 ) SYSOT1=SYSOT1+WORDOT(I)*I1
      SYSIN2=0
      SYSOT2=0
      DO 14 I=32,62
        I1=2**(I-32)
        SYSIN2=SYSIN2+WORDIN(I)*I1
14   IF( FLAGS(16) .EQ. 1 ) SYSOT2=SYSOT2+WORDOT(I)*I1
      DO 20 I=1,15000
20   SEMANT(I)=BLANK
      DO 25 J=1,5
      DO 25 I=1,3000
25   POINTR(I,J)=0
      IF( FLAGS(14) .GE. 1 ) CALL APTCTT
      IF( FLAGS(15) .EQ. 1 ) CALL UPDATE
      IF( FLAGS(13) .EQ. 1 ) CALL APTTTC
C   DECK 7
      LINES=0
      PAGE=0
      L=0
      RNPREV=0
40   READ (INTAPE,1002) RULENO,SUBRUL,WORD,OD
1002 FORMAT(14,1X,12,1X,62I1,2X,A8)
      IF( RULENO .EQ. 9999) GO TO 100
      I1=0
      DO 41 I=1,31
41   I1=I1+WORD(I)*2**(I-1)
      I2=0
      DO 42 I=32,59
42   I2=I2+WORD(I)*2**(I-32)
      IF( FLAGS(16) .EQ. 0 ) GO TO 65
      IN1=0

```

```

      IF( (AND(I1,SYSIN1).NE.0) .OR. (AND(I2,SYSIN2).NE.0) ) IN1=1      MAIN0610
      IF( (AND(I1,SYSOT1).NE.0) .OR. (AND(I2,SYSOT2).NE.0) ) IN1=IN1+2  MAIN0620
      IF( (IN1.LE.0) .OR. (IN1.GE.3) ) GO TO 40      MAIN0630
      GO TO 68                                       MAIN0640
65 IF( (AND(I1,SYSIN1).EQ.0) .AND. (AND(I2,SYSIN2).EQ.0) ) GO TO 40    MAIN0650
68 IF( RULENO .EQ. RNPREV ) GO TO 70               MAIN0660
      IFLG=1                                         MAIN0670
      POINTR(RULENO,2)=L+1                          MAIN0680
      RNPREV=RULENO                                 MAIN0690
      POINTR(RULENO,3)=0                            MAIN0700
70 L=L+1                                           MAIN0710
      SYSTEM(L)=SUBRUL                             MAIN0720
      POINTR(RULENO,3)=POINTR(RULENO,3)+1          MAIN0730
      IF( FLAGS(16) .NE. 1 ) GO TO 78              MAIN0740
      L=L+1                                         MAIN0750
      SYSTEM(L)=IN1                                MAIN0760
      GO TO 40                                     MAIN0770
78 IF( FLAGS(7) .NE. 1 ) GO TO 40                 MAIN0780
      IF( LINES .GT. 0 ) GO TO 80                  MAIN0790
      PAGE=PAGE+1                                  MAIN0800
      WRITE (6,1003) DATE,PAGE,WORDIN,(SYSNAM(INM,1),INM=1,9)          MAIN0810
1003 FORMAT('1NUMERICAL CONTROL META-LANGUAGE SYSTEM FLAGS',5X,A8,5X,   MAIN0820
1'PAGE-',I2/'0IN SYSTEMS-',62I1/' SYSTEM NAME-', 9A8/'ORULE SUBRULE'  MAIN0830
2',62X,'UPDATED')
      LINES=54                                     MAIN0840
80 LINES=LINES-1                                  MAIN0850
      IF( IFLG .GT. 0 ) GO TO 90                   MAIN0860
      WRITE (6,1004) SUBRUL,WORD,0D                MAIN0870
1004 FORMAT(' ',7X,I2,4X,62I1,2X,A8)              MAIN0880
      GO TO 40                                     MAIN0890
90 IFLG=0                                          MAIN0900
      WRITE (6,1005) RULENO,SUBRUL,WORD,0D         MAIN0910
1005 FORMAT(' ',I4,I5,4X,62I1,2X,A8)              MAIN0920
      GO TO 40                                     MAIN0930
C      APT SUBSETS AND MODULES                    MAIN0940
0100 IF( FLAGS(6) .EQ. 1 ) CALL MODULE             MAIN0950
C      DECK 8                                       MAIN0960
      LINES=0                                       MAIN0970
      ISUB=FLAGS(8)                                MAIN0980
      NOPREV=0                                      MAIN0990
      PAGE=0                                        MAIN1000
      NOSYS=L                                      MAIN1010
      NTRY=0                                        MAIN1020
110 READ (INTAPE,1014) RULENO,CONT,SUBRUL,DESC,0D MAIN1030
1014 FORMAT(I4,A1,I2,1X,63A1,1X,A8)              MAIN1040
      IF( RULENO .EQ. 9999 ) GO TO 150             MAIN1050
      IF( POINTR(RULENO,2) .LE. 0 ) GO TO 110      MAIN1060
      IF( RULENO .EQ. NOPREV ) IF( FLAGS(8)-1 ) 110,110,126 MAIN1070
      SRPREV=-10                                   MAIN1080
      DO 115 I=1,63                                MAIN1090
      I1=64-I                                       MAIN1100
115 IF( DESC(I1) .NE. BLNK1 ) GO TO 120           MAIN1110
120 POINTR(RULENO,4)=NTRY+1                       MAIN1120
      POINTR(RULENO,5)=I1                         MAIN1130
      DO 125 I=1,I1                                MAIN1140
      NTRY=NTRY+1                                  MAIN1150
125 SEMANT(NTRY)=DESC(I)                          MAIN1160
      SEMDAT(RULENO)=0D                           MAIN1170
      IF( FLAGS(16) .EQ. 1 ) GO TO 146             MAIN1180
      IF( FLAGS(8) .LT. 1 ) GO TO 146             MAIN1190
126 IF( LINES .GT. 0 ) GO TO 130                  MAIN1200
      PAGE=PAGE+1                                  MAIN1210
      WRITE (6,1006) SEMNAM(ISUB),DATE,PAGE,WORDIN, MAIN1220
1(SYSNAM(INM,1),INM=1,7)                          MAIN1230
1006 FORMAT('1NUMERICAL CONTROL META-LANGUAGE ',A8, '5',5X,A8,5X,   MAIN1240
1'PAGE-',I2/'0IN SYSTEMS-',62I1/' SYSTEM NAME-', 9A8/'ORULE SUBRULE'  MAIN1250
2',62X,'UPDATED')
      LINES=54                                     MAIN1260
                                           MAIN1270
                                           MAIN1280

```

130 IF(FLAGS(8) .EQ. 1) GO TO 145	MAIN1290
IF(RULENO .NE. NOPREV) GO TO 145	MAIN1300
IF(SUBRUL .NE. SRPREV) GO TO 135	MAIN1310
IF(J .LE. 0) GO TO 110	MAIN1320
WRITE (6,1017) CONT,DESC,OD	MAIN1330
1017 FORMAT(' ',9X,A1,1X,63A1,A8)	MAIN1340
GO TO 148	MAIN1350
0135 J=0	MAIN1360
J1=POINTR(RULENO,2)	MAIN1370
J2=POINTR(RULENO,3)+J1-1	MAIN1380
DO 140 K=J1,J2	MAIN1390
0140 IF(SYSTEM(K) .EQ. SUBRUL) GO TO 142	MAIN1400
GO TO 110	MAIN1410
0142 J=1	MAIN1420
WRITE (6,1018) SUBRUL,DESC,OD	MAIN1430
1018 FORMAT(' ',4X,I5,2X,63A1,A8)	MAIN1440
GO TO 147	MAIN1450
145 WRITE (6,1007) RULENO,SUBRUL,DESC,OD	MAIN1460
1007 FORMAT(' ',I4,I5,2X,63A1,A8)	MAIN1470
146 NOPREV=RULENO	MAIN1480
0147 SRPREV=SUBRUL	MAIN1490
0148 LINES=LINES-1	MAIN1500
GO TO 110	MAIN1510
C DECK 9	MAIN1520
150 LINES=0	MAIN1530
IDTSYN=0	MAIN1540
IF(FLAGS(16) .NE. 1) GO TO 158	MAIN1550
CALL COMSYS	MAIN1560
REWIND INTAPE	MAIN1570
GO TO 10	MAIN1580
158 PAGE=0	MAIN1590
RNPREV=0	MAIN1600
J=0	MAIN1610
RNPRV2=0	MAIN1620
160 READ (INTAPE,1008) RULENO,CONT,SUBRUL,IT,OD	MAIN1630
1008 FORMAT(I4,I1,I2,1X,15I4,4X,A8)	MAIN1640
IF(RULENO .EQ. 9999) GO TO 250	MAIN1650
IF(POINTR(RULENO,2) .LE. 0) GO TO 160	MAIN1660
J1=POINTR(RULENO,2)	MAIN1670
J2=POINTR(RULENO,3)+J1-1	MAIN1680
DO 170 K=J1,J2	MAIN1690
170 IF(SYSTEM(K) .EQ. SUBRUL) GO TO 180	MAIN1700
GO TO 160	MAIN1710
180 IF(CONT .NE. 0) GO TO 200	MAIN1720
IF(RULENO .EQ. RNPREV) GO TO 190	MAIN1730
L=J+1	MAIN1740
RNPREV=RULENO	MAIN1750
POINTR(RULENO,1)=L	MAIN1760
SYNTAX(L)=0	MAIN1770
J=L	MAIN1780
190 SYNTAX(J+1)=SUBRUL	MAIN1790
IDTSYN=IDTSYN+1	MAIN1800
SYNTAX(J+2)=IDTSYN	MAIN1810
SYNDAT(IDTSYN)=OD	MAIN1820
SYNTAX(L)=SYNTAX(L)+1	MAIN1830
M=J+3	MAIN1840
SYNTAX(M)=0	MAIN1850
J=M	MAIN1860
200 DO 210 K=1,15	MAIN1870
IF(IT(K) .EQ. 0) GO TO 220	MAIN1880
J=J+1	MAIN1890
210 SYNTAX(J)=IT(K)	MAIN1900
220 K=K-1	MAIN1910
SYNTAX(M)=SYNTAX(M)+K	MAIN1920
IF(FLAGS(9) .LE. 0) GO TO 160	MAIN1930
IF(LINES .GT. 0) GO TO 222	MAIN1940
PAGE=PAGE+1	MAIN1950
WRITE (6,1010) DATE,PAGE,WORDIN,(SYSNAM(INM,1),INM=1,9)	MAIN1960

```

1010 FORMAT('INUMERICAL CONTROL META-LANGUAGE SYNTAX',5X,A8,5X,'PAGE-',MAIN1970
112/'OIN SYSTEMS-',62I1/' SYSTEM NAME-', 9A8/'ORULE SUBRULE',62X,MAIN1980
2'UPDATED')
LINES=54
222 CNTP=0
223 LINES=LINES-1
      KP=MINO(K,12)
      DO 224 K1=1,KP
      CALL INTHOL (IT(K1),AR(K1),LEN)
224 IF( LEN .LT. 5 ) CALL INSERT (AR(K1),1,5-LEN,BLANK)
      IF( KP.EQ. 12 ) GO TO 227
      K1=KP+1
      DO 226 K2=K1,12
226 AR(K2)=BLANK
227 IF( CNTP .LE. 0 ) GO TO 228
      WRITE (6,1011) AR
      GO TO 238
228 IF( CONT .EQ. 0 ) GO TO 230
229 WRITE (6,1011) AR,OD
1011 FORMAT(14X,12A5,1X,A8)
      GO TO 238
230 IF( RULENO .EQ. RNPRV2 ) GO TO 232
      RNPRV2=RULENO
      WRITE (6,1012) RULENO,SUBRUL,AR,OD
1012 FORMAT(' ',14,15,4X,12A5,1X,A8)
      GO TO 238
232 WRITE (6,1013) SUBRUL,AR,OD
1013 FORMAT(' ',4X,15,4X,12A5,1X,A8)
238 IF( K .LE. 12 ) GO TO 160
      CNTP=1
      DO 240 K1=13,K
240 IT(K1-12)=IT(K1)
      K=K-12
      GO TO 223
250 REWIND INTAPE
      NOSYN=J
      DO 280 I=1,1999
      IF( POINTR(I,1) .LE. 0 ) GO TO 280
      I1=POINTR(I,1)
      IF( SYNTAX(I1) .NE. 1 ) GO TO 280
      IF( SYNTAX(I1+3) .NE. 1 ) GO TO 280
      IF( SYNTAX(I1+4) .NE. 128 ) GO TO 280
      DO 270 K=1,1999
      IF( POINTR(K,1) .LE. 0 ) GO TO 270
      K1=POINTR(K,1)
      IF( SYNTAX(K1) .LE. 0 ) GO TO 270
      K2=SYNTAX(K1)
      K5=K1
      DO 260 K3=1,K2
      K4=K5+4
      K5=K4+SYNTAX(K4-1)-1
      DO 260 K6=K4,K5
      IF( I .NE. SYNTAX(K6) ) GO TO 260
      SYNTAX(K6)=-128
260 CONTINUE
270 CONTINUE
      POINTR(I,1)=0
280 CONTINUE
      IF( FLAGS(10) .EQ. 1 ) CALL XREFER
      IF( FLAGS(11) .GT. 0 ) CALL SHORT
      IF( FLAGS(12) .GT. 0 ) CALL LONG
      WRITE (6,1009) NOSYN,NOSYS,NTRY,IDTSYN
1009 FORMAT('ENTRIES IN SYNTAX TABLE =',I5/'ENTRIES IN SYSTEMS TABLE
1=' ,I5/'ENTRIES IN SEMANTICS TABLE =',I5/'ENTRIES IN SYNTAX DATE
2TABLE =',I5)
      GO TO 10
300 STOP
      END

```

```

BLOCK DATA
IMPLICIT INTEGER*4 (A-Z)
COMMON /COM/ ASCII(128),DATE,SEMANT(25000),FLAGS(20),SYNTAX(20000)
1,POINTR(3000,5),SYSTEM(5000),INTAPE,WORDIN(62),WORDOT(62)
COMMON /OTHER/ SYSNAM(9,2)
COMMON /CONTNO/ CONTIN(37)
LOGICAL*1 CONTIN
LOGICAL*1 SEMANT,ASCII
REAL*8 DATE,SYSNAM
INTEGER*2 SYSTEM,SUBRUL
DATA CONTIN/'0 123456789ABCDEFGHIJKLMNQPQRSTUVWXYZ' /
DATA ASCII/128H $ '()*+,-./0123456789:; = ABCDEFGHIJKLMNQPQRSTUVWXYZ |
2 /
END

```

BLOC0010
BLOC0020
BLOC0030
BLOC0040
BLOC0050
BLOC0060
BLOC0070
BLOC0080
BLOC0090
BLOC0100
BLOC0110
BLOC0120
BLOC0130
BLOC0140
BLOC0150

```

SUBROUTINE APTCTT
IMPLICIT INTEGER*4 (A-Z)
COMMON /COM/ ASCII(128),DATE,SEMANT(25000),FLAGS(20),SYNTAX(20000)
1,POINTR(3000,5),SYSTEM(5000),INTAPE,WORDIN(62),WORDOT(62)
COMMON /OTHER/ SYSNAM(9,2)
LOGICAL*1 SEMANT,ASCII
REAL*8 DATE,SYSNAM
INTEGER*2 SYSTEM,SUBRUL
REAL*8 A(10)
PAGE=0
LINES=0
N=0
INP=5
INTAPE=1
IF( FLAGS(14) .NE. 2 ) GO TO 10
FLAGS(15)=0
INP=1
INTAPE=2
10 READ (INP,1000,END=20) A(1),I,(A(J),J=2,10)
1000 FORMAT(A7,I1,9A8)
N=N+1
WRITE (INTAPE,1000) A(1),I,(A(J),J=2,9),DATE
IF( LINES .GT. 0 ) GO TO 15
LINES=55
PAGE=PAGE+1
WRITE (6,1002) DATE,PAGE
1002 FORMAT('NUMERICAL CONTROL META-LANGUAGE CARD TO TAPE',5X,A8,5X,
1'PAGE-',I3/)
15 WRITE (6,1001) A(1),I,(A(J),J=2,10)
1001 FORMAT(' ',A7,I1,9A8)
IF( I .EQ. 9 ) GO TO 20
LINES=LINES-1
GO TO 10
20 WRITE (6,1003) N
1003 FORMAT('NUMBER OF CARDS =',I5)
END FILE INTAPE
REWIND INTAPE
IF( FLAGS(14) .EQ. 2 ) REWIND INP
RETURN
END

```

APTC0010
APTC0020
APTC0030
APTC0040
APTC0050
APTC0060
APTC0070
APTC0080
APTC0090
APTC0100
APTC0110
APTC0120
APTC0130
APTC0140
APTC0150
APTC0160
APTC0170
APTC0180
APTC0190
APTC0200
APTC0210
APTC0220
APTC0230
APTC0240
APTC0250
APTC0260
APTC0270
APTC0280
APTC0290
APTC0300
APTC0310
APTC0320
APTC0330
APTC0340
APTC0350
APTC0360
APTC0370
APTC0380
APTC0390
APTC0400

```

SUBROUTINE APTTTC
IMPLICIT INTEGER*4 (A-Z)
COMMON /COM/ ASCII(128),DATE,SEMANT(25000),FLAGS(20),SYNTAX(20000)
1,POINTR(3000,5),SYSTEM(5000),INTAPE,WORDIN(62),WORDOT(62)
COMMON /OTHER/ SYSNAM(9,2)
COMMON /WHICH/ IPUNW(5)
LOGICAL*1 SEMANT,ASCII
REAL*8 DATE,SYSNAM
INTEGER*2 SYSTEM,SUBRUL
REAL*8 A(9),B
LINES=0

```

APTT0010
APTT0020
APTT0030
APTT0040
APTT0050
APTT0060
APTT0070
APTT0080
APTT0090
APTT0100
APTT0110

PAGE=0	APTT0120
N=0	APTT0130
ISW=0	APTT0140
IF(IPUNW(1)+IPUNW(2) .GT. 0) ISW=1	APTT0150
10 READ (INTAPE,1001,END=20) I,A,J,B	APTT0160
1001 FORMAT(I4,8A8,A3,I1,A8)	APTT0170
IF(ISW .LE. 0) GO TO 100	APTT0180
IF(IPUNW(1) .GT. 1) GO TO 10	APTT0190
IF(IPUNW(2) .LT. 1) GO TO 10	APTT0200
IF(J .LE. 6) GO TO 10	APTT0210
IF(IPUNW(J-4) .LE. 0) GO TO 10	APTT0220
100 N=N+1	APTT0230
WRITE (7,1001) I,A,J,B	APTT0240
IF(LINES .GT. 0) GO TO 15	APTT0250
LINES=55	APTT0260
PAGE=PAGE+1	APTT0270
WRITE (6,1000) DATE,PAGE	APTT0280
1000 FORMAT('1NUMERICAL CONTROL META-LANGUAGE TAPE TO CARD',5X,A8,5X,	APTT0290
1'PAGE-',I3/)	APTT0300
15 WRITE (6,1003) I,A,J,B	APTT0310
1003 FORMAT(' ',I4,8A8,A3,I1,A8)	APTT0320
LINES=LINES-1	APTT0330
GO TO 10	APTT0340
20 WRITE (6,1002) N	APTT0350
1002 FORMAT('0NUMBER OF CARDS =',I5)	APTT0360
REWIND INTAPE	APTT0370
RETURN	APTT0380
END	APTT0390
SUBROUTINE COMSYS	COMS0010
C FLAGS(16)=1	COMS0020
IMPLICIT INTEGER*4 (A-Z)	COMS0030
COMMON /CCM/ ASCII(128),DATE,SEMANT(25000),FLAGS(20),SYNTAX(20000)	COMS0040
1,POINTR(3000,5),SYSTEM(5000),INTAPE,WORDIN(62),WORDOT(62)	COMS0050
COMMON /OTHER/ SYSNAM(9,2)	COMS0060
LOGICAL*1 SEMANT,ASCII	COMS0070
REAL*8 DATE,SYSNAM	COMS0080
INTEGER*2 SYSTEM,SUBRUL	COMS0090
LOGICAL*1 CHAR(2)/' X'/,CRO(2)	COMS0100
LINES=0	COMS0110
PAGE=0	COMS0120
DO 50 IR=1,3000	COMS0130
IF(POINTR(IR,2) .LE. 0) GO TO 50	COMS0140
INSR=POINTR(IR,3)	COMS0150
I1=POINTR(IR,2)	COMS0160
J1=POINTR(IR,4)	COMS0170
JNUM=MINO(POINTR(IR,5),57)	COMS0180
J2=.I1+JNUM-1	COMS0190
LSR=I1-2	COMS0200
DO 40 IS=1,INSR	COMS0210
LSR=LSR+2	COMS0220
ISR=SYSTEM(LSR)	COMS0230
CRO(1)=CHAR(1)	COMS0240
CRO(2)=CHAR(1)	COMS0250
ISR1=SYSTEM(LSR+1)	COMS0260
CRO(ISR1)=CHAR(2)	COMS0270
IF(LINES .GT. 0) GO TO 20	COMS0280
PAGE=PAGE+1	COMS0290
LINES=50	COMS0300
WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(I,1),I=1,9),WORDOT,	COMS0310
1(SYSNAM(I,2),I=1,9)	COMS0320
1000 FORMAT('1NUMERICAL CONTROL META-LANGUAGE COMPARISON',5X,A8,5X,	COMS0330
1'PAGE-',I2/'0SYSTEMS 1 -',62I1/' SYSTEM NAME-',9A8/' SYSTEMS 2 -'	COMS0340
2,62I1/' SYSTEM NAME-',9A8/ 'ORULE SUBRULE 1 2 (X MEANS IN SYSTE	COMS0350
3M)')	COMS0360
0020 LINES=LINES-1	COMS0370
IF(IS .NE. 1) GO TO 30	COMS0380
WRITE (6,1001) IR,ISR,CRO,(SEMANT(J),J=J1,J2)	COMS0390

```

1001 FORMAT(' ',I4,I5,4X,A1,2X,A1,5X,57A1)
GO TO 40
0030 WRITE (6,1002) ISR,CRO
1002 FORMAT(' ',4X,I5,4X,A1,2X,A1)
0040 CONTINUE
0050 CONTINUE
RETURN
END

SUBROUTINE LONG
IMPLICIT INTEGER*4 (A-Z)
COMMON /COM/ ASCII(128),DATE,SEMANT(25000),FLAGS(20),SYNTAX(20000)
1,POINTR(3000,5),SYSTEM(5000),INTAPE,WORDIN(62),WORDOT(62)
COMMON /OTHER/ SYSNAM(9,2)
LOGICAL*1 SEMANT,ASCII
REAL*8 DATE,SYSNAM
INTEGER*2 SYSTEM,SUBRUL
REAL*4 TSR(3)/' ' IS OR NOT/'
LOGICAL*1 LB/'<'/,RB/'>'/,BLK1/' '/,STR(5),STRING(118)
NCT=FLAGS(12)*50+17
LINES=0
PAGE=0
DO 150 IR=1,3000
IF( POINTR(IR,1) .LE. 0 ) GO TO 150
I1=POINTR(IR,1)
I4=POINTR(IR,4)
I5=POINTR(IR,5)+I4-1
NSR=SYNTAX(I1)
IF( NSR .LE. 0 ) GO TO 150
IF( LINES .GT. 3 ) GO TO 10
LINES=52
PAGE=PAGE+1
WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(INM,1),INM=1,9)
1000 FORMAT('1NUMERICAL CONTROL META-LANGUAGE',5X,A8,5X,'PAGE-',I2/
'1OIN SYSTEMS-',62I1/' SYSTEM NAME-',9A8)
10 IF( I5 .GE. I4 ) GO TO 20
WRITE (6,1001) IR,RB
1001 FORMAT('0<',I4,1X,64A1)
GO TO 30
20 WRITE (6,1001) IR,(SEMANT(I),I=I4,I5),RB
30 J2=I1
LINES=LINES-2
ISW2=1
DO 130 I=1,NSR
J1=J2+4
J2=J1+SYNTAX(J1-1)-1
CONT=0
IF( SYNTAX(J1-3) .LT. 0 ) ISW2=3
ISRN=IABS(SYNTAX(J1-3))
NSC=0
DO 100 J=J1,J2
J3=SYNTAX(J)
CALL INTHOL (J3,STR,L)
IF( J3 .LE. 0 ) GO TO 34
K4=POINTR(J3,4)
K5=POINTR(J3,5)
GO TO 38
34 K5=1
38 IF( NSC+K5+7 .LE. NCT ) GO TO 50
IF( LINES .GT. 0 ) GO TO 39
PAGE=PAGE+1
WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(INM,1),INM=1,9)
LINES=52
39 LINES=LINES-1
IF( CONT .LE. 0 ) GO TO 40
WRITE (6,1002) (STRING(K),K=1,NSC)
1002 FORMAT(13X,118A1)
GO TO 45
40 WRITE (6,1003) TSR(ISW2),ISRN,(STRING(K),K=1,NSC)

```

```

COMS0400
COMS0410
COMS0420
COMS0430
COMS0440
COMS0450
COMS0460
COMS0470

LONG0010
LONG0020
LONG0030
LONG0040
LONG0050
LONG0060
LONG0070
LONG0080
LONG0090
LONG0100
LONG0110
LONG0120
LONG0130
LONG0140
LONG0150
LONG0160
LONG0170
LONG0180
LONG0190
LONG0200
LONG0210
LONG0220
LONG0230
LONG0240
LONG0250
LONG0260
LONG0270
LONG0280
LONG0290
LONG0300
LONG0310
LONG0320
LONG0330
LONG0340
LONG0350
LONG0360
LONG0370
LONG0380
LONG0390
LONG0400
LONG0410
LONG0420
LONG0430
LONG0440
LONG0450
LONG0460
LONG0470
LONG0480
LONG0490
LONG0500
LONG0510
LONG0520
LONG0530
LONG0540
LONG0550
LONG0560
LONG0570
LONG0580
LONG0590
LONG0600

```



```

1003 FORMAT(4X,A4,'(',I2,')',1X,118A1) LONG0610
      CONT=1 LONG0620
45 NSC=0 LONG0630
50 NSC=NSC+1 LONG0640
      STRING(NSC)=LB LONG0650
      DO 60 K=1,L LONG0660
      NSC=NSC+1 LONG0670
60 STRING(NSC)=STR(K) LONG0680
      NSC=NSC+1 LONG0690
      STRING(NSC)=BLK1 LONG0700
      IF( J3 .GT. 0 ) GO TO 70 LONG0710
      J3=-J3 LONG0720
      NSC=NSC+1 LONG0730
      STRING(NSC)=ASCII(J3) LONG0740
      GO TO 80 LONG0750
70 IF( K5 .LE. 0 ) GO TO 80 LONG0760
      K5=K5+K4-1 LONG0770
      DO 75 K=K4,K5 LONG0780
      NSC=NSC+1 LONG0790
75 STRING(NSC)=SEMANT(K) LONG0800
80 NSC=NSC+1 LONG0810
      STRING(NSC)=RB LONG0820
100 CONTINUE LONG0830
      IF( IINFS .GT. 0 ) GO TO 105 LONG0840
      PAGE=PAGE+1 LONG0850
      WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(INM,1),INM=1,9) LONG0860
      LINES=52 LONG0870
105 LINES=LINES-1 LONG0880
      IF( CONT .LE. 0 ) GO TO 110 LONG0890
      WRITE (6,1002) (STRING(K),K=1,NSC) LONG0900
      GO TO 130 LONG0910
110 WRITE (6,1003) TSR(ISW2),ISRN,(STRING(K),K=1,NSC) LONG0920
130 ISW2=2 LONG0930
150 CONTINUE LONG0940
      RETURN LONG0950
      END LONG0960

      SUBROUTINE MODULE MODU0010
      IMPLICIT INTEGER*4 (A-Z) MODU0020
      COMMON /COM/ ASCII(128),DATE,SEMANT(25000),FLAGS(20),SYNTAX(20000) MODU0030
1,POINTR(3000,5),SYSTEM(5000),INTAPE,WORDIN(62),WORDOT(62) MODU0040
      COMMON /OTHER/ SYSNAM(9,2) MODU0050
      LOGICAL*1 SEMANT,ASCII MODU0060
      REAL*8 DATE,SYSNAM,OD MODU0070
      INTEGER*2 SYSTEM,SUBRUL MODU0080
      DIMENSION WORD(62),IREC(7) MODU0090
      INTEGER*4 BLANK/' ',SUBNO(5)/' 1 2 3 4 5 '/, MODU0100
1MODNO(25)/'1.2 1.3 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 3.1 3.2 3.3 3.4 MODU0110
2 3.5 3.6 3.7 3.8 3.9 4.1 4.2 4.3 4.4 4.5 4.6 '/ MODU0120
      REWIND INTAPE MODU0130
      LINES=0 MODU0140
      PAGE=0 MODU0150
0100 READ (INTAPE,1002) RULENO,SUBRUL,WORD,OD MODU0160
1002 FORMAT(I4,1X,I2,1X,62I1,2X,A8) MODU0170
      IF( RULENO .EQ. 9999 ) GO TO 200 MODU0180
      IF( RULENO .GE. 2000 ) GO TO 100 MODU0190
      IF( POINTR(RULENO,2) .LE. 0 ) GO TO 100 MODU0200
      J1=POINTR(RULENO,2) MODU0210
      J2=POINTR(RULENO,3)+J1-1 MODU0220
      DO 110 K=J1,J2 MODU0230
0110 IF( SYSTEM(K) .EQ. SUBRUL ) GO TO 120 MODU0240
      GO TO 100 MODU0250
0120 IF( LINES .GT. 0 ) GO TO 130 MODU0260
      PAGE=PAGE+1 MODU0270
      LINES=53 MODU0280
      NOPREV=0 MODU0290
      WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(I,1),I=1,9) MODU0300

```

```

1000 FORMAT('1NUMERICAL CONTROL META-LANGUAGE SUBSETS AND MODULES', MODU0310
      15X,A8,5X,'PAGE-',I2/'0IN SYSTEM-',62I1/' SYSTEM NAME-',9A8/ MODU0320
      2'ORULE SUBRULE SUBSETS MODULAR-FEATURE'/32X,'NO. MIN.SUBS MODU0330
      3ET') MODU0340
0130 LINES=LINES-1 MODU0350
      DO 140 I=5,9 MODU0360
      IREC(I-4)=BLANK MODU0370
0140 IF( WORD(I) .GT. 0 ) IREC(I-4)=SUBNO(I-4) MODU0380
      IREC(6)=BLANK MODU0390
      IREC(7)=BLANK MODU0400
      DO 150 I=25,49 MODU0410
      IF( WORD(I) .LE. 0 ) GO TO 150 MODU0420
      IREC(6)=MCDNO(I-24) MODU0430
      GO TO 160 MODU0440
0150 CONTINUE MODU0450
      DO 163 I=19,23 MODU0460
      IF( WORD(I) .LE. 0 ) GO TO 163 MODU0470
      IREC(7)=SUBNO(I-18) MODU0480
      GO TO 166 MODU0490
      163 CONTINUE MODU0500
0166 IF(RULENO .EQ. NOPREV ) GO TO 170 MODU0510
      NOPREV=RULENO MODU0520
      WRITE (6,1003) RULENO,SUBRUL,IREC MODU0530
1003 FORMAT(' ',I4,I5,6X,5A2,6X,A3,4X,A2) MODU0540
      GO TO 100 MODU0550
0170 WRITE (6,1004) SUBRUL,IREC MODU0560
1004 FORMAT(' ',4X,I5,6X,5A2,6X,A3,4X,A2) MODU0570
      GO TO 100 MODU0580
0200 RETURN MODU0590
      END MODU0600

```

```

SUBROUTINE SHORT SHOR0010
IMPLICIT INTEGER*4 (A-Z) SHOR0020
COMMON /CGM/ ASCII(128),DATE,SEMANT(25000),FLAGS(20),SYNTAX(20000) SHOR0030
1,POINTR(3000,5),SYSTEM(5000),INTAPE,WORDIN(62),WORDOT(62) SHOR0040
COMMON /OTHER/ SYSNAM(9,2) SHOR0050
COMMON /DATES/ SEMDAT(3000),SYNDAT(8000) SHOR0060
REAL*8 SEMDAT,SYNDAT SHOR0070
LOGICAL*1 SEMANT,ASCII SHOR0080
REAL*8 DATE,SYSNAM SHOR0090
INTEGER*2 SYSTEM,SUBRUL SHOR0100
LOGICAL*1 LB/'<'/,RB/'>'/,STR(5),STRING(58)/58*' ' / SHOR0110
LOGICAL*1 BLK1/' ' / SHOR0120
REAL*4 TSR(3)/' IS OR NOT' / SHOR0130
NCT=58 SHOR0140
LINES=0 SHOR0150
PAGE=0 SHOR0160
DO 200 IR=1,3000 SHOR0170
IF( POINTR(IR,1) .LE. 0 ) GO TO 200 SHOR0180
I1=POINTR(IR,1) SHOR0190
NSR=SYNTAX(I1) SHOR0200
IF( NSR .LE. 0 ) GO TO 200 SHOR0210
I4=POINTR(IR,4) SHOR0220
I5=MINO(60,POINTR(IR,5)) SHOR0230
I6=61-I5 SHOR0240
I5=I5+I4-1 SHOR0250
IF( LINES .GT. 3 ) GO TO 10 SHOR0260
LINES=52 SHOR0270
PAGE=PAGE+1 SHOR0280
WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(I,1),I=1,9) SHOR0290
1000 FORMAT('1NUMERICAL CONTROL META-LANGUAGE (ABBREVIATED)',5X,A8,5X, SHOR0300
      1'PAGE-',I2/'0IN SYSTEMS-',62I1/' SYSTEM NAME-',9A8/'ORULE SUBRULE SHOR0310
      2',60X,'UPDATED'//) SHOR0320
      10 IF( I5 .GE. I4 ) GO TO 20 SHOR0330
      WRITE (6,1001) IR,RB,(BLK1,I=1,61),SEMDAT(IR) SHOR0340
1001 FORMAT('0<',I4,1X,62A1,3X,A8) SHOR0350
      GO TO 30 SHOR0360
      20 WRITE (6,1001) IR,(SEMANT(I),I=I4,I5),RB,(BLK1,I=1,I6),SEMDAT(IR) SHOR0370
      30 LINES=LINES-2 SHOR0380

```

```

J2=11
ISW2=1
DO 170 I=1,NSR
J1=J2+4
J2=J1+SYNTAX(J1-1)-1
II=SYNTAX(J1-2)
CONT=0
IF( SYNTAX(J1-3) .LT. 0 ) ISW2=3
ISRN=IABS(SYNTAX(J1-3))
NSC=0
DO 150 J=J1,J2
J3=SYNTAX(J)
IF( J3 .GT. 2031 ) GO TO 110
IF( J3 .LE. 0 ) GO TO 100
C POSITIVE ELEMENT <2032
35 CALL INTROL (J3,STR,L)
K4=POINTR(J3,4)
K5=MIN0(50,POINTR(J3,5))
IF( NSC+K5+7 .LE. NCT ) GO TO 50
IF( LINES .GT. 0 ) GO TO 38
PAGE=PAGE+1
LINES=52
WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(INM,1),INM=1,9)
38 LINES=LINES-1
IF( CONT .LC. 0 ) GO TO 40
WRITE (6,1002) (STRING(K),K=1,NSC)
1002 FORMAT(13X,58A1)
GO TO 41
40 WRITE (6,1003) TSR(ISW2),ISRN,(STRING(K),K=1,58),SYNDAT(II)
1003 FORMAT(4X,A4,'(',I2,')',1X,58A1,A9)
CONT=1
41 DO 42 K=1,58
42 STRING(K)=BLK1
45 NSC=0
50 NSC=NSC+1
STRING(NSC)=LB
DO 60 K=1,L
NSC=NSC+1
60 STRING(NSC)=STR(K)
NSC=NSC+1
STRING(NSC)=BLK1
IF( K5 .LE. 0 ) GO TO 80
K5=K5+K4-1
DO 75 K=K4,K5
NSC=NSC+1
75 STRING(NSC)=SEMANT(K)
80 NSC=NSC+1
STRING(NSC)=RR
GO TO 150
C NEGATIVE ELEMENT
100 IF( J3 .EQ. -128 ) GO TO 150
NSC=NSC+1
J3=-J3
STRING(NSC)=ASCII(J3)
GO TO 150
C POSITIVE ELEMENT >2000
110 L1=POINTR(J3,1)
NSR1=SYNTAX(L1)
IF( NSR1 .LE. 0 ) GO TO 35
L2=L1+4
L3=L2+SYNTAX(L1+3)-1
DO 120 K=L2,L3
IF( SYNTAX(K) .LT. 0 ) GO TO 120
IF( SYNTAX(K) .GT. 2128 ) GO TO 35
IF( SYNTAX(K) .LT. 2032 ) GO TO 35
120 CONTINUE
IF( NSC+L3-L2+1 .LE. NCT ) GO TO 126
IF( LINES .GT. 0 ) GO TO 122
PAGE=PAGE+1

```

```

SHOR0390
SHOR0400
SHOR0410
SHOR0420
SHOR0430
SHOR0440
SHOR0450
SHOR0460
SHOR0470
SHOR0480
SHOR0490
SHOR0500
SHOR0510
SHOR0520
SHOR0530
SHOR0540
SHOR0550
SHOR0560
SHOR0570
SHOR0580
SHOR0590
SHOR0600
SHOR0610
SHOR0620
SHOR0630
SHOR0640
SHOR0650
SHOR0660
SHOR0670
SHOR0680
SHOR0690
SHOR0700
SHOR0710
SHOR0720
SHOR0730
SHOR0740
SHOR0750
SHOR0760
SHOR0770
SHOR0780
SHOR0790
SHOR0800
SHOR0810
SHOR0820
SHOR0830
SHOR0840
SHOR0850
SHOR0860
SHOR0870
SHOR0880
SHOR0890
SHOR0900
SHOR0910
SHOR0920
SHOR0930
SHOR0940
SHOR0950
SHOR0960
SHOR0970
SHOR0980
SHOR0990
SHOR1000
SHOR1010
SHOR1020
SHOR1030
SHOR1040
SHOR1050
SHOR1060
SHOR1070

```

```

      LINES=52
      WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(INM,1),INM=1,9)
122  LINES=LINES-1
      IF( CONT .LE. 0 ) GO TO 123
      WRITE (6,1002) (STRING(K),K=1,NSC)
      GO TO 124
123  WRITE (6,1003) TSR(ISW2),ISRN,(STRING(K),K=1,58),SYNDAT(II)
      CONT=1
124  DO 125 K=1,58
125  STRING(K)=BLK1
      NSC=0
126  DO 130 K=L2,L3
      NSC=NSC+1
      K1=SYNTAX(K)
      IF( K1 .LT. 0 ) GO TO 128
      K1=POINTR(K1,1)
      K1=SYNTAX(K1+4)
      IF( K1 .GT. 0 ) K1=-128
128  K1=-K1
130  STRING(NSC)=ASCII(K1)
150  CONTINUE
      IF( LINES .GT. 0 ) GO TO 155
      PAGE=PAGE+1
      LINES=52
      WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(INM,1),INM=1,9)
155  LINES=LINES-1
      IF( CONT .LE. 0 ) GO TO 160
      WRITE (6,1002) (STRING(K),K=1,NSC)
      GO TO 162
160  WRITE (6,1003) TSR(ISW2),ISRN,(STRING(K),K=1,58),SYNDAT(II)
162  DO 165 K=1,58
165  STRING(K)=BLK1
170  ISW2=2
200  CONTINUE
      RETURN
      END

```

```

SHOR1080
SHOR1090
SHOR1100
SHOR1110
SHOR1120
SHOR1130
SHOR1140
SHOR1150
SHOR1160
SHOR1170
SHOR1180
SHOR1190
SHOR1200
SHOR1210
SHOR1220
SHOR1230
SHOR1240
SHOR1250
SHOR1260
SHOR1270
SHOR1280
SHOR1290
SHOR1300
SHOR1310
SHOR1320
SHOR1330
SHOR1340
SHOR1350
SHOR1360
SHOR1370
SHOR1380
SHOR1390
SHOR1400
SHOR1410
SHOR1420
SHOR1430

```

```

      SUBROUTINE UPDATE
      IMPLICIT INTEGER*4 (A-Z)
      COMMON /CGM/ ASCII(128),DATE,SEMANT(25000),FLAGS(20),SYNTAX(20000)
1  ,POINTR(3000,5),SYSTEM(5000),INTAPE,WORDIN(62),WORDOT(62)
      COMMON /OTHER/ SYSNAM(9,2)
      COMMON /CCNTNO/ CONTIN(37)
      LOGICAL*1 SEMANT,ASCII,PA(72),PAT(72),BLK/' '/
      LOGICAL*1 CONTIN,I2,IZ1
      REAL*8 DATE,SYSNAM
      INTEGER*2 SYSTEM,SUBRUL
      REAL*8 A(9),AT(9)
      EQUIVALENCE (PA(1),A(1)),(PAT(1),AT(1))
      IDKP=0
      REWIND 1
      REWIND 2
      ISW1=1
      NC=0
      PAGE=0
      LINES=0
10  READ (5,1003,END=200) IR,IZ1,ISR,IFL,(A(I),I=1,8),IDK,A(9)
1003 FORMAT(I4,A1,I2,I1,7A8,A7,I1,A8)
      IF( LINES .GT. 0 ) GO TO 15
      PAGE=PAGE+1
      LINES=56
      WRITE (6,999) DATE,PAGE
999  FORMAT('1NUMERICAL CONTROL META-LANGUAGE UPDATE CARDS',A10,5X,
1  'PAGE-',I2/)
      LINES=LINES-1
      WRITE (6,1001) IR,IZ1,ISR,IFL,(A(I),I=1,8),IDK,A(9)
1001 FORMAT(' ',I4,A1,I2,I1,7A8,A7,I1,A8)
      IF( IFL .EQ. 9 ) GO TO 200
      DO 16 I=1,37

```

```

UPDA0010
UPDA0020
UPDA0030
UPDA0040
UPDA0050
UPDA0060
UPDA0070
UPDA0080
UPDA0090
UPDA0100
UPDA0110
UPDA0120
UPDA0130
UPDA0140
UPDA0150
UPDA0160
UPDA0170
UPDA0180
UPDA0190
UPDA0200
UPDA0210
UPDA0220
UPDA0230
UPDA0240
UPDA0250
UPDA0260
UPDA0270
UPDA0280
UPDA0290
UPDA0300
UPDA0310
UPDA0320

```

```

16 IF( IZ1.EQ. CONTIN(I) ) GO TO 17
17 IC=MAX0(0,I-2)
   NC=NC+1
   IR=IABS(IR)
   IF( IDKP - IDK ) 60,20,50
20 IF( IRP - IR ) 70,30,50
30 IF( ISRP - IABS(ISR) ) 80,40,50
40 IF( ICP - IC ) 90,50,50
50 WRITE (6,1002)
1002 FORMAT(' ***ABOVE CARD IGNORED ***')
   LINES=LINES-1
   GO TO 10
60 IDKP=IDK
70 IRP=IR
80 ISRP=IABS(ISR)
90 ICP=IC
100 GO TO (120,110),ISW1
110 ISW1=1
   GO TO 125
120 READ (1,1000,END=230) IRT, IZ,ISRT,IFLT,(AT(I),I=1,8),IDKT,AT(9)
1000 FORMAT(I4,A1,I2,I1,7A8,A7,I1,A8)
   DO 122 I=1,37
122 IF( IZ .EQ. CONTIN(I) ) GO TO 123
123 ICT=MAX0(0,I-2)
125 IF( IDK - IDKT ) 170,130,160
130 IF( IR - IRT ) 170,140,160
140 IF( IABS(ISR) - IABS(ISRT) ) 170,150,160
150 IF( IC - ICT ) 170,180,160
160 WRITE (2,1000)IRT,CONTIN(ICT+2),ISRT,IFLT,(AT(I),I=1,8),IDKT,AT(9)
   GO TO 120
170 ISW1=2
   IF( IDK .EQ. 7 ) PA(1)=CONTIN(3)
   IF( IFL-1 ) 190,10,190
180 IF( IFL .EQ. 1 ) GO TO 10
   IF( IDK .GT. 7 ) GO TO 190
   PA(1)=CONTIN(3)
   DO 185 I=2,62
   IF( PA(I) .NE. BLK ) GO TO 185
   PA(I)=PAT(I)
185 CONTINUE
190 WRITE (2,1000) IR,CONTIN(IC+2),ISR,IFL,(A(I),I=1,8),IDK,DATE
   GO TO 10
200 GO TO (210,220),ISW1
210 READ (1,1000,END=350) IRT,IZ,ISRT,IFLT,(AT(I),I=1,8),IDKT,AT(9)
220 WRITE (2,1000)IRT,IZ,ISRT,IFLT,(AT(I),I=1,8),IDKT,AT(9)
   GO TO 210
230 WRITE (2,1000)IR,CONTIN(IC+2),ISR,IFL,(A(I),I=1,8),IDK,DATE
240 READ (5,1003,END=350) IR,IZ,ISR,IFL,(A(I),I=1,8),IDK,A(9)
   IF( LINES .GT. 0 ) GO TO 243
   PAGE=PAGE+1
   WRITE (6,999) DATE,PAGE
   LINES=56
243 LINES=LINES-1
   WRITE (6,1001) IR,IZ,ISR,IFL,(A(I),I=1,8),IDK,A(9)
   IF( IFL .EQ. 9 ) GO TO 350
   DO 245 I=1,37
245 IF( IZ .EQ. CONTIN(I) ) GO TO 246
246 IC=MAX0(0,I-2)
   NC=NC+1
   IF( IDKP - IDK ) 290,250,280
250 IF( IRP - IR ) 300,260,280
260 IF( ISRP - IABS(ISR) ) 310,270,280
270 IF( ICP - IC ) 320,280,280
280 WRITE (6,1002)
   LINES=LINES-1
   GO TO 240
290 IDKP=IDK
300 IRP=IR
310 ISRP=IABS(ISR)

```

UPDA0330
 UPDA0340
 UPDA0350
 UPDA0360
 UPDA0370
 UPDA0380
 UPDA0390
 UPDA0400
 UPDA0410
 UPDA0420
 UPDA0430
 UPDA0440
 UPDA0450
 UPDA0460
 UPDA0470
 UPDA0480
 UPDA0490
 UPDA0500
 UPDA0510
 UPDA0520
 UPDA0530
 UPDA0540
 UPDA0550
 UPDA0560
 UPDA0570
 UPDA0580
 UPDA0590
 UPDA0600
 UPDA0610
 UPDA0620
 UPDA0630
 UPDA0640
 UPDA0650
 UPDA0660
 UPDA0670
 UPDA0680
 UPDA0690
 UPDA0700
 UPDA0710
 UPDA0720
 UPDA0730
 UPDA0740
 UPDA0750
 UPDA0760
 UPDA0770
 UPDA0780
 UPDA0790
 UPDA0800
 UPDA0810
 UPDA0820
 UPDA0830
 UPDA0840
 UPDA0850
 UPDA0860
 UPDA0870
 UPDA0880
 UPDA0890
 UPDA0900
 UPDA0910
 UPDA0920
 UPDA0930
 UPDA0940
 UPDA0950
 UPDA0960
 UPDA0970
 UPDA0980
 UPDA0990
 UPDA1000
 UPDA1010

320	ICP=IC	UPDA1020
	IF(IDK .EQ. 7) PA(1)=CONTIN(3)	UPDA1030
	GO TO 230	UPDA1040
350	END FILE 2	UPDA1050
	REWIND 1	UPDA1060
	REWIND 2	UPDA1070
	INTAPE=2	UPDA1080
	WRITE (6,1004) NC	UPDA1090
1004	FORMAT('NUMBER OF UPDATE CARDS =',I4)	UPDA1100
	RETURN	UPDA1110
	END	UPDA1120
	SUBROUTINE XREFER	XREF0010
C	FLAGS(10)=1	XREF0020
	IMPLICIT INTEGER*4 (A-Z)	XREF0030
	COMMON /COM/ ASCII(128),DATE,SEMANT(25000),FLAGS(20),SYNTAX(20000)	XREF0040
	1,POINTR(3000,5),SYSTEM(5000),INTAPE,WORDIN(62),WORDOT(62)	XREF0050
	COMMON /OTHER/ SYSNAM(9,2)	XREF0060
	LOGICAL*1 SEMANT,ASCII	XREF0070
	REAL*8 DATE,SYSNAM	XREF0080
	INTEGER*2 SYSTEM,SUBRUL	XREF0090
	REAL*8 T(9),BLANK/' ' //,AIR/' ' //	XREF0100
	DATA AST/'* ' //,BLK/' ' //	XREF0110
	PAGE=0	XREF0120
	LINES=0	XREF0130
	DO 200 IR=1,3000	XREF0140
	IDID=0	XREF0150
	ASTFLG=BLK	XREF0160
	IF(POINTR(IR,1) .LE. 0) ASTFLG=AST	XREF0170
0010	CALL INTHOL (IR,AIR,LEN)	XREF0180
	IF(LEN .LT. 4) CALL INSERT (AIR,1,4-LEN,BLANK)	XREF0190
	NRU=0	XREF0200
	DO 110 IR1=1,2199	XREF0210
	IF(POINTR(IR1,1) .LE. 0) GO TO 110	XREF0220
	I1=POINTR(IR1,1)	XREF0230
	NSR=SYNTAX(I1)	XREF0240
	IF(NSR .LE. 0) GO TO 110	XREF0250
	J=I1	XREF0260
	DO 100 N=1,NSR	XREF0270
	J1=J+4	XREF0280
	J2=J1+SYNTAX(J+3)-1	XREF0290
	DO 50 I=J1,J2	XREF0300
0050	IF(SYNTAX(I) .EQ. IR) GO TO 60	XREF0310
	GO TO 100	XREF0320
0060	NRU=NRU+1	XREF0330
	CALL INTHOL (IR1,T(NRU),LEN)	XREF0340
	IF(LEN .LT. 4) CALL INSERT (T(NRU),1,4-LEN,BLANK)	XREF0350
	CALL INTHOL (SYNTAX(J+1),A,LEN)	XREF0360
	IF(LEN .LT. 2) CALL INSERT (A,1,2-LEN,BLANK)	XREF0370
	CALL INSERT (A,1,1,'-')	XREF0380
	CALL INSERT (T(NRU),5,3,A)	XREF0390
	IF(NRU .LT. 9) GO TO 100	XREF0400
	IF(LINES .GT. 0) GO TO 70	XREF0410
	PAGE=PAGE+1	XREF0420
	WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(I,1),I=1,9)	XREF0430
1000	FORMAT('NUMERICAL CONTROL RULE CROSS-REFERENCE',5X,A8,5X,	XREF0440
	1'PAGE-',12/'OIN SYSTEMS-',62I1/' SYSTEM NAME-',9A8/	XREF0450
	2'ORULE RULE-SUBRULE')	XREF0460
	LINES=50	XREF0470
0070	LINES=LINES-1	XREF0480
	WRITE (6,1001) AIR,ASTFLG,(T(I),I=1,NRU)	XREF0490
1001	FORMAT(' ',A4,A1,A7,8(1X,A7))	XREF0500
	IDID=1	XREF0510
	AIR=BLANK	XREF0520
	NRU=0	XREF0530
0100	J=J2	XREF0540
0110	CONTINUE	XREF0550
	IF(IDID .GT. 0) GO TO 200	XREF0560
	IF(NRU .GT. 0) GO TO 115	XREF0570

IF(POINTR(IR,1) .LE. 0) GO TO 200	XREF0580
T(1)=BLANK	XREF0590
115 IF(LINES .GT. 0) GO TO 120	XREF0600
LINES=50	XREF0610
PAGE=PAGE+1	XREF0620
WRITE (6,1000) DATE,PAGE,WORDIN,(SYSNAM(I,1),I=1,9)	XREF0630
0120 LINES=LINES-1	XREF0640
WRITE (6,1001) AIR,ASTFLG,(T(I),I=1,NRU)	XREF0650
AIR=BLANK	XREF0660
0200 CONTINUE	XREF0670
RETURN	XREF0680
END	XREF0690
000010 IDENTIFICATION DIVISION.	APTS0010
000020 PROGRAM-ID. 'APTSTR'.	APTS0020
000030 ENVIRONMENT DIVISION.	APTS0030
000040 CONFIGURATION SECTION.	APTS0040
000050 SOURCE-COMPUTER. IBM-360 F50.	APTS0050
000060 OBJECT-COMPUTER. IBM-360 F50.	APTS0060
000100 DATA DIVISION.	APTS0070
000110 WORKING-STORAGE SECTION.	APTS0080
000120 77 K PICTURE S9(8) COMPUTATIONAL.	APTS0090
000130 77 L PICTURE S9(8) COMPUTATIONAL.	APTS0100
000140 01 J-REPT PICTURE -(9)9.	APTS0110
000150 01 J-REDEF REDEFINES J-REPT.	APTS0120
000160 02 J-CHAR PICTURE X OCCURS 10.	APTS0130
000170 01 STRNEW.	APTS0140
000180 02 CH-NEW PICTURE X OCCURS 72.	APTS0150
000190 LINKAGE SECTION.	APTS0160
000200 77 J PICTURE S9(8) COMPUTATIONAL.	APTS0170
000210 77 N PICTURE S9(8) COMPUTATIONAL.	APTS0180
000220 01 Q.	APTS0190
000230 02 P PICTURE X OCCURS 64.	APTS0200
000240 01 STROLD.	APTS0210
000250 02 CH-OLD PICTURE X OCCURS 72.	APTS0220
000260 01 RSTROLD REDEFINES STROLD.	APTS0230
000270 02 WORD1 PICTURE X(4).	APTS0240
000280 01 CHAR.	APTS0250
000290 02 CHAR1 PICTURE X.	APTS0260
000300 02 CHAR2 PICTURE XXX.	APTS0270
001000 PROCEDURE DIVISION.	APTS0280
001010 INSERTS.	APTS0290
001020 ENTER LINKAGE.	APTS0300
001030 ENTRY 'INSERT' USING STROLD, J, N, Q.	APTS0310
001040 ENTER COBOL.	APTS0320
001050 COMPUTE L = 0.	APTS0330
001060 IF J < 1 COMPUTE J = 1, MOVE ' ' TO CH-OLD (1),	APTS0340
001070 COMPUTE TALLY = 0, GO TO NEW-STR.	APTS0350
001080 EXAMINE STROLD TALLYING UNTIL FIRST ' '.	APTS0360
001090 IF J = 1 GO TO NEW-STR.	APTS0370
001100 IF J > (TALLY + 1) COMPUTE J = TALLY + 1.	APTS0380
001110 PERFORM MOVE1 VARYING K FROM 1 BY 1 UNTIL K > (J - 1).	APTS0390
001120 NEW-STR.	APTS0400
001130 PERFORM MOVE2 VARYING K FROM 1 BY 1 UNTIL K > N.	APTS0410
001140 PERFORM MOVE1 VARYING K FROM J BY 1 UNTIL K > (TALLY + 1).	APTS0420
001150 PERFORM MOVE4 VARYING K FROM 1 BY 1 UNTIL K > L.	APTS0430
001160 ENTER LINKAGE.	APTS0440
001170 RETURN.	APTS0450
001180 ENTER COBOL.	APTS0460
002010 INTBCOLS.	APTS0470
002020 ENTER LINKAGE.	APTS0480
002030 ENTRY 'INTHOL' USING J, STROLD, N.	APTS0490
002040 ENTER COBOL.	APTS0500
002050 MOVE ' ' TO WORD1.	APTS0510
002060 MOVE J TO J REPT.	APTS0520
002070 EXAMINE J-REPT TALLYING LEADING ' '.	APTS0530
002080 COMPUTE N = 10 - TALLY.	APTS0540
002090 PERFORM MOVE3 VARYING K FROM 1 BY 1 UNTIL K > N.	APTS0550
002100 MOVE ' ' TO CH-OLD (K).	APTS0560

002110	ENTER LINKAGE.	APTS0570
002120	RETURN.	APTS0580
002130	ENTER COBOL.	APTS0590
002500	COMPARES.	APTS0600
002510	ENTER LINKAGE.	APTS0610
002520	ENTRY 'CMPAR' USING STROLD, Q, N, J.	APTS0620
002530	ENTER COBOL.	APTS0630
002540	PERFORM BYTE-COMPARE THRU BC-END VARYING K FROM 1 BY 1	APTS0640
002550	UNTIL K > N.	APTS0650
002560	COMPUTE J = 0.	APTS0660
002570	COMPARES-END.	APTS0670
002580	ENTER LINKAGE.	APTS0680
002590	RETURN.	APTS0690
002600	ENTER COBOL.	APTS0700
003000	FETCHES.	APTS0710
003010	ENTER LINKAGE.	APTS0720
003020	ENTRY 'FETCH' USING STROLD, J, CHAR.	APTS0730
003030	ENTER COBOL.	APTS0740
003040	MOVE CH-OLD (J) TO CHAR1.	APTS0750
003050	MOVE ' ' TO CHAR2.	APTS0760
003060	ENTER LINKAGE.	APTS0770
003070	RETURN.	APTS0780
003080	ENTER COBOL.	APTS0790
004000	MOVE1.	APTS0800
004010	COMPUTE L = L + 1.	APTS0810
004020	MOVE CH-OLD (K) TO CH-NEW (L).	APTS0820
004030	MOVE2.	APTS0830
004040	COMPUTE L = L + 1.	APTS0840
004050	MOVE P (K) TO CH-NEW (L).	APTS0850
004060	MOVE3.	APTS0860
004070	COMPUTE TALLY = TALLY + 1.	APTS0870
004080	MOVE J-CHAR (TALLY) TO CH-OLD (K).	APTS0880
004090	MOVE4.	APTS0890
004100	MOVE CH-NEW (K) TO CH-OLD (K).	APTS0900
004110	BYTE-COMPARE.	APTS0910
004120	IF CH-OLD (K) = P (K) GO TO BC-END ELSE	APTS0920
004130	COMPUTE J = 1, GO TO COMPARES-END.	APTS0930
004140	BC-END. EXIT.	APTS0940
005000	FINISH. STOP RUN.	APTS0950