84000450

CONTROL DATA CORPORATION

SIMSCRIPT II.5 INSTANT MANUAL

CDC® OPERATING SYSTEMS
NOS
NOS/BE
SCOPE 2

84000450

CONTROL DATA CORPORATION

SIMSCRIPT II.5 INSTANT MANUAL

CDC® OPERATING SYSTEMS
NOS
NOS/BE
SCOPE 2

REVISION RECORD

REVISION	DESCRIPTION
A	Initial printing
(12-1-77)	
B	Revised to add the EXTENDED ENTITIES ARRAYS INCLUDE
(9-25-78)	language statement, the MASK.F function mnemonic, and
	the MOVE.F routine. Other minor changes and corrections
	made.

Publication No.
84000450

REVISION LETTERS I, O, Q AND X ARE NOT USED

Printed in the United States
of America

Address comments concerning this
manual to:

CONTROL DATA CORPORATION
Publications and Graphics Division
Data Services Publications
P. O. Box 0 HQC02C
Minneapolis, MN 55440

PREFACE

This instant outlines the SIMSCRIPT II.5 programming language for the CONTROL DATA computer systems. More detailed information on the SIMSCRIPT II.5 language can be obtained from the following manuals:

P.J. Kiviat, R. Villanueva, H.M. Markowitz
SIMSCRIPT II.5 Programming Language
(CACI)

SIMSCRIPT II.5 Reference Handbook
(CACI)

SIMSCRIPT II.5 User's Manual—CONTROL DATA computer systems (CDC)
(CACI)

CONTENTS

Preface iii

SIMSCRIPT II.5 ELEMENTS 1
- Names 1
- Constants 1
- Variables 1

SOURCE LANGUAGE 2
- Notation 2
- Symbols 3
- Metavariables 4
- Language Statements 7
- Attributes and routines required for set statements 22
- Statistical keywords for ACCUMULATE statement 23
- Statistical keywords for TALLY statement 24
- Preamble statement recap 25

SYSTEM DEFINED NAMES 26
- System defined variables 26
- System defined constants 28
- System defined functions 29
- System defined routines 37
- Generated variables, attributes and routines 38

SIMII5 COMPILATION CONTROL CARD 41

SIMII5 EXECUTION CONTROL CARD 43

SIMII5 CHARACTER SETS 44

SIMSCRIPT II.5 ELEMENTS

NAMES		Examples
Combination of letters, digits and periods arranged as follows: – contains at least one letter or – contains digits and at least two non-terminal periods. – cannot be broken across card boundaries.		X WAITING.LINE 1B 1.2.3 1.5E15

CONSTANTS		Examples
Integer	Combination of up to 18 digits	2 347 314159265
Real	Combination of up to 15 digits containing at least one period	3.14 1. .759
String	Sequence of characters enclosed within ".	"ABCD" ". . . ."
Text	Sequence of characters enclosed within |.	|TEXT IS:|

VARIABLES	
Integer	Range $-(2^{59}-1)$ to $2^{59}-1$. As a result of multiplication or division or conversion from real to integer, or integer to real, maximum value is $2^{48}-1$.
Integer packed	. Bit packed (n-m) $1 \leqslant n \leqslant m \leqslant 60$. Field packed (n/m) m = 2, 3, 4, 5, 6, 10, 12, 15 n = 1 to m . Intra-packed (*/m) m = 2, 3, 4, 5, 6, 10, 12, 15
Real	Range 10^{-293} to 10^{+322} (approximately 15 significant digits)
Alpha	Up to 10 characters for a replacement statement.

NOTATION

The notation employed in describing SIMSCRIPT II.5 is a combination of conventions used in several computer programming language descriptions. In the following pages:

1. Words in capital letters are statement *keywords*.
2. *Primitives* shown in italics are basic language constructs. They are:

   name

   integer *i*

   number *n*

   string *s*

   text *t*

3. A *metavariable* denotes an occurrence of an element of the type represented by the metavariable symbol shown in italics.
4. A *statement* is a combination of keywords, primitives, and metavariables that follows a certain pattern, called the syntax of the statement.
5. Brackets [ ] and braces { } denote choices. When brackets appear, a choice *may* be made from the options indicated. When braces appear, a choice *must* be made. The items available for selection appear in a vertical list within the brackets or braces. When a choice can be repeated, a symbol (or symbols) that must separate the items in the list of choices is written at the upper right-hand corner of the brackets or braces. For example, if a choice appears as $\left\{\begin{matrix}A\\B\end{matrix}\right\}'$ the sequence A,A,B,A, . . ., B might be selected. The choice represented by $\{A\}'$ is logically equivalent to A [,A] [,A] . . . [,A].
6. The null character ※ is used to indicate that no symbol need separate the items in a list of choices. An example of $\left\{\begin{matrix}A\\B\end{matrix}\right\}^{※}$ might be AABABB . . . A. The choice represented by $\{A\}^{※}$ is logically equivalent to A [A] [A] . . . [A].
7. A list separator symbol can itself be complex, involving choices and repetitions, as in $\left\{\begin{matrix}A\\B\end{matrix}\right\}\left\{\begin{matrix}AND\\OR\end{matrix}\right\}$ an instance of which might be A AND B OR A OR B.
8. Plural *keywords* ending in S such as VARIABLES and LINES, can be written in singular form as VARIABLE or LINE when called for by the grammar of a statement.

SYMBOLS

character = { *letter* | *special character* | *digit* | *blank* | *period* }

letter = { A B C D E F G H I J K L M N O P Q R S T U V W X Y Z }

special character = { * ** , ( ) < > | - + = " ' : ; / ! ¢ ? % _ & ¬ $ @ # }

digit = { 0 1 2 3 4 5 6 7 8 9 }

blank = empty space

period = .

METAVARIABLES

arithmetic expression e =

$$\left[\begin{matrix}+\\-\end{matrix}\right]\left\{\begin{matrix}(e)\\ n\\ r\\ s\\ l\\ [\$]v\end{matrix}\right\}\left\{\begin{matrix}+\\-\\*\\**\\/\end{matrix}\right\}$$

---

comma c =

$$\left\{\begin{matrix},\\ ,\text{AND}\\ \text{AND}\end{matrix}\right\}$$

---

for phrase *for* =

$$\text{FOR}\left\{\begin{matrix}name\ \left\{\begin{matrix}\text{BACK FROM}\\ =\end{matrix}\right\}e\ \text{TO}\ e\ [\text{BY}\ e]\\ \left\{\begin{matrix}\text{EACH}\\ \text{EVERY}\\ \text{ALL}\end{matrix}\right\}\left\{\begin{matrix}name\ [\text{CALLED}\ name]\\ name\left[\left\{\begin{matrix}\text{FROM}\\ \text{AFTER}\end{matrix}\right\}e\right]\left\{\begin{matrix}\text{OF}\\ \text{IN}\\ \text{ON}\\ \text{AT}\end{matrix}\right\}v\ [\text{IN REVERSE ORDER}]\end{matrix}\right.\\ \text{EACH}\ name\ \text{IN THE DICTIONARY}\end{matrix}\right\}[,][\text{TO}]^{*}$$

---

format f_1 =

$$\left\{\begin{matrix}\text{B}\,e\\ \text{S}\,e\\ /\end{matrix}\right\}$$

---

format f_2 =

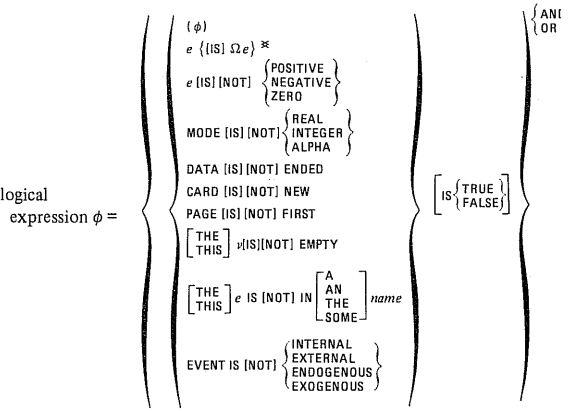
$$\left\{\begin{matrix}f_1\\ i\ \text{C}\,e\\ i\ \text{I}\,e\\ i\ \text{A}\,e\\ i\ \text{D}\,(e,e)\\ i\ \text{E}\,(e,e)\\ i\ \text{T}\,e\end{matrix}\right\}$$

---

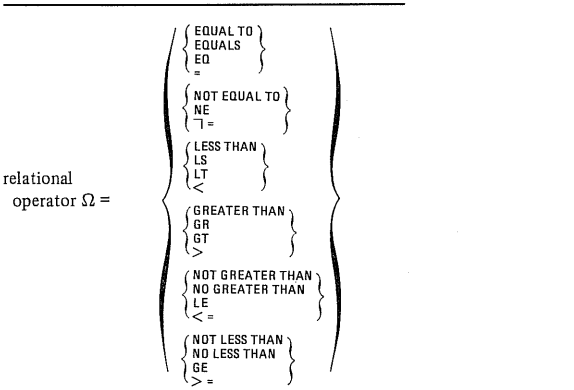
label l =

$$\left\{\begin{matrix}name\\ n\end{matrix}\right\}$$

logical expression ϕ =



relational operator Ω =



routine r = *'name'*

selection clause sc = {WITH / [EXCEPT] WHEN / UNLESS} ϕ [,]

termination clause tc = { WHILE / UNTIL } ϕ [,]

---

variable v = *name* [({ $\{e\}'$ [, $\{*\}'$] / $\{*\}'$ })]

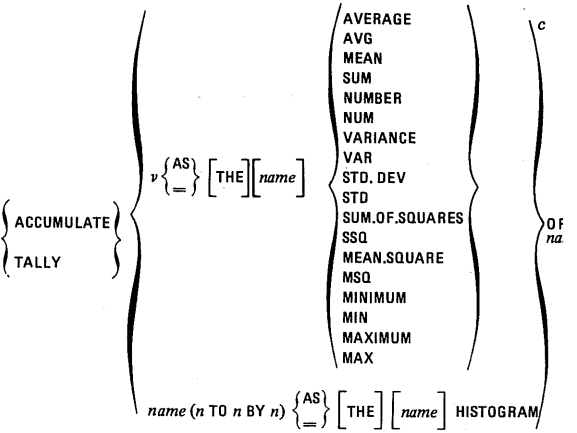
---

word w = { *t* / *name* / *n* / *special character* / *s* }

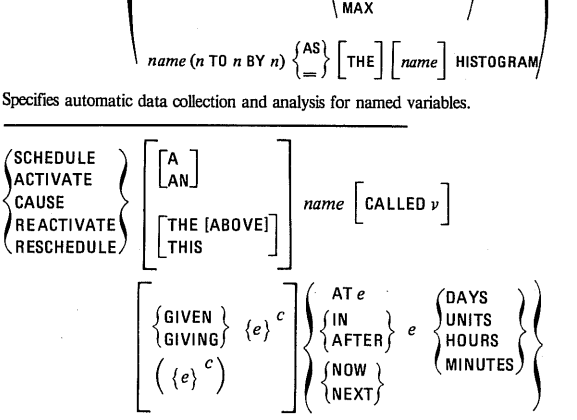
Words must be separated from each other by one or more blanks unless they are special characters.

Periods (.) are ignored between words and at the end of statements.

Comments can be inserted between any two words in a program by enclosing them in quote marks (") formed by two consecutive apostrophes. The right-hand set of quotes is not necessary if the comment is the last item on a card.



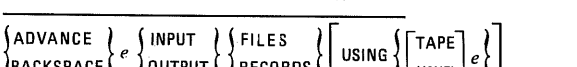
Specifies automatic data collection and analysis for named variables.

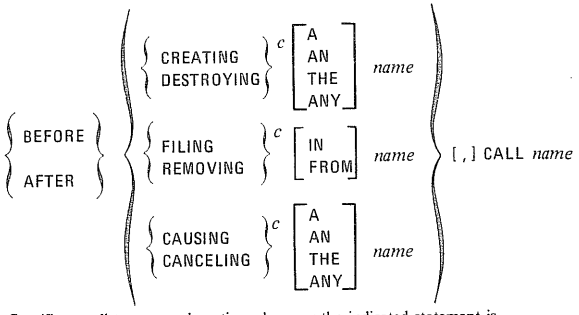


Files an event notice in the events set according to its time.

ADD *e* TO *v*

Adds the value of *e* to the value of the variable *v*.

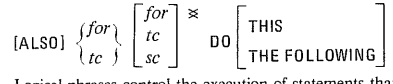




Specifies a call to a named routine whenever the indicated statement is executed. Inputs to the routines are:

	BEFORE	AFTER
CREATE	Not allowed	Entity identifier
DESTROY	Entity identifier	Not allowed
CAUSE	Entity identifier, time	Entity identifier, time
CANCEL	Entity identifier	Entity identifier
FILE	Entity identifier, subscripts	Entity identifier, subscripts
REMOVE	Entity identifier, subscripts	Entity identifier, subscripts

---



Logical phrases control the execution of statements that follow them. When more than one statement is to be controlled, the word DO precedes them. Multiple control phrases terminating control on the same LOOP statement are preceded by the word ALSO.

ALWAYS
REGARDLESS

Synonyms indicating the transfer point of the false condition of a preceding IF statement.

BACKSPACE — See ADVANCE

BEFORE — See AFTER

BEGIN HEADING

Marks the beginning of a heading block within a report section.

BEGIN REPORT [ON A NEW PAGE] [PRINTING *for* IN GROUPS OF *i* [PER PAGE]]

Marks the beginning of a report section.

BREAK *name* TIES { BY { HIGH / LOW } *name* } *c* THEN

Establishes a priority order within an event class.

{ PERFORM / CALL / NOW } *name* [{ THE / THIS / GIVEN / GIVING } {*e*}^*c* / ({*e*}^*c*)] [YIELDING {*v*}^*c*]

Calls a routine used as a procedure. Both input GIVEN and output YIELDING argument lists are optional.

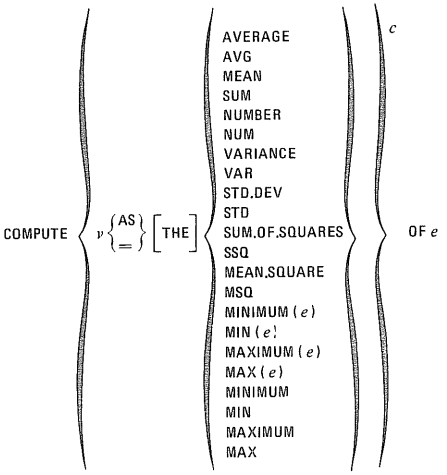
CANCEL [THE [ABOVE] / THIS] *name* [CALLED *v*]

Removes a scheduled event notice from the event set.

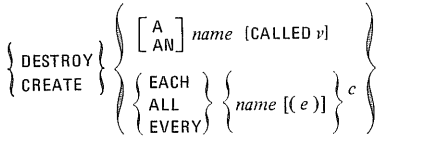
CAUSE — See ACTIVATE

{ CLOSE / ENDFILE } [TAPE / UNIT] *e*

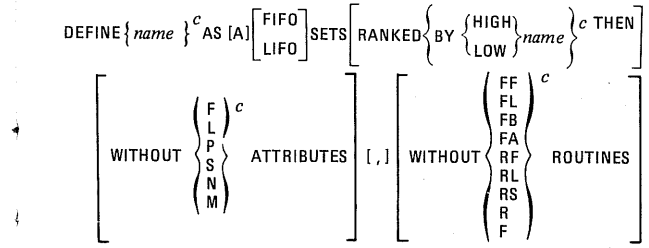
Writes an end-of-file mark on an output device.



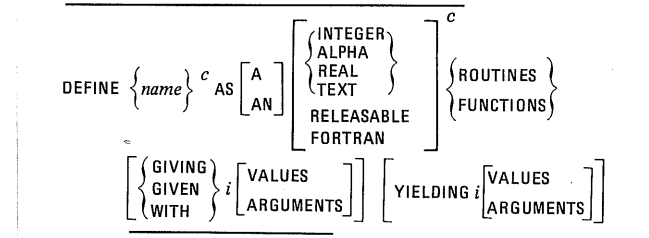
Must be controlled by a logical control phrase. Computes the indicated statistics of the expression *e* after the LOOP statement if the control is over a DO. . .LOOP block.

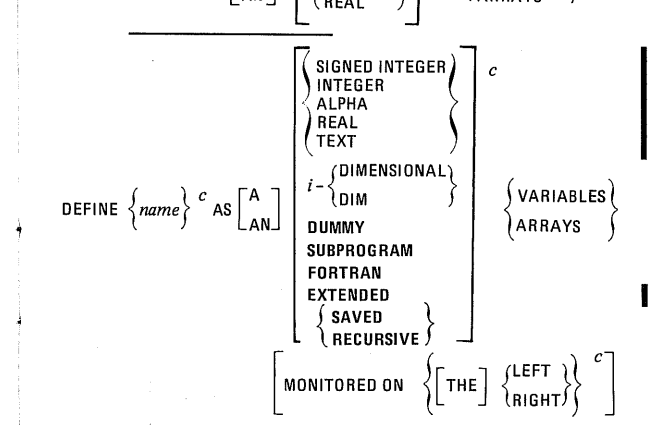


A control statement.



Defines set ranking, owner and member attributes and generated set processing routines.





Define properties of global and local variables, and routines.

DEFINE *w* TO MEAN $\{w\}^{\infty}$

Instructs the compiler to substitute the words (up to the end of the card on which the statement appears) following the keyword MEAN for the indicated word in all subsequent statements, before they are compiled.

DESTROY — See CREATE

DO — See ALSO

{ELSE / OTHERWISE}

Synonyms indicating the transfer point of the false condition of a preceding IF statement.

END

Marks the end of a program preamble, routine, report section, and heading block of a report section.

ENTER WITH *v*

Used to transfer a "right-hand" value to a left-handed function.

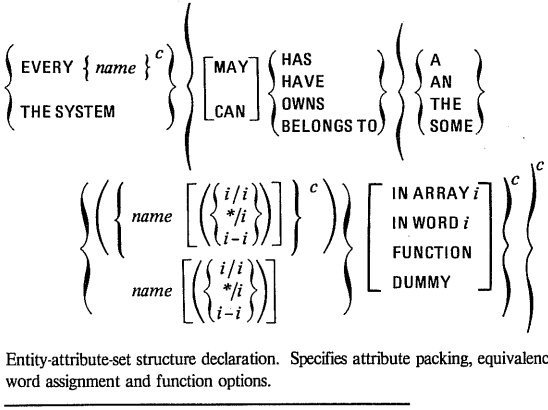
{UPON / EVENT} *name* [{ THE / THIS / GIVEN / GIVING } {*name*}^{c} | ({*name*}^{c})] [SAVING THE EVENT NOTICE]

Event declaration. Unless SAVED, an event notice is destroyed before an event routine is executed.

EVENT NOTICES [INCLUDE {*name*}^{c}]

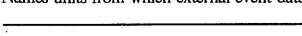
{ EVERY {*name*}^{c} / THE SYSTEM } { [MAY / CAN] { HAS / HAVE } { { A / AN / THE / SOME } *name* } }

RANDOM [STEP / LINEAR] { VARIABLES / ARRAYS } [IN ARRAY *i* / IN WORD *i*] }^{c} }^{c}

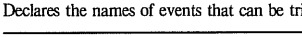


Entity-attribute-set structure declaration. Specifies attribute packing, equivalence, word assignment and function options.

Defines extended memory data structures.



Names units from which external event data will be read.



Declares the names of events that can be triggered externally.

Files an entity in a set.

Must be controlled by a logical control phrase, but cannot be within a DO...LOOP block. The optional IF phrase directs control after the control phrase has been completed, depending upon the "success" of the FIND.

FOR — See ALSO and for metavariable

$$\begin{bmatrix}\text{LEFT}\\ \text{RIGHT}\end{bmatrix}\begin{Bmatrix}\text{ROUTINE}\\ \text{FUNCTION}\\ \text{SUBROUTINE}\end{Bmatrix}\begin{bmatrix}\text{TO}\\ \text{FOR}\end{bmatrix}\textit{name}\begin{bmatrix}\begin{Bmatrix}\text{THE}\\ \text{THIS}\\ \text{GIVING}\\ \text{GIVEN}\end{Bmatrix}\left\{\textit{name}\right\}^{c}\\ \\ (\{\textit{name}\}^{c})\end{bmatrix}$$

$$\left[\text{YIELDING }\{\textit{name}\}^{c}\right]$$

Subprogram declaration. Routines used as functions only have GIVEN arguments. If LEFT or RIGHT are not stated, RIGHT is implied.

$$\begin{Bmatrix}\text{GENERATE}\\ \text{INHIBIT}\end{Bmatrix}\ \text{LIST ROUTINES}$$

Controls the generation of LIST ATTRIBUTES routines for entities. By default, no LIST routines are generated.

$$\text{GO [TO]}\ \begin{Bmatrix}\text{'}l\ [(e)]\text{'}\\ l\ [(e)]\end{Bmatrix}$$

Transfers control to the indicated label.

$$\text{GO [TO]}\begin{Bmatrix}l\\ \text{'}l\text{'}\end{Bmatrix}\begin{Bmatrix},\\ \text{OR}\end{Bmatrix}\ \text{PER}\ e$$

Transfers control to the n th label in the label list according to the integer value of the transfer expression *e*.

HERE — A control statement

IF Ø[,] — See Ø metavariable

INTERRUPT *name* [CALLED *v*]

$$\text{JUMP}\ \begin{Bmatrix}\text{AHEAD}\\ \text{BACK}\end{Bmatrix}$$

A control statement.

'*l* [(*i*)]'

A statement label identifies a transfer point.

LAST COLUMN { **IS** / **=** } *i*

Characters beyond column *i* are ignored on subsequent cards.

LEAVE — A control statement.

LET *v* = *e*

Assigns the value of *e* to the variable *v*.
If *v* is integer and *e* is real, the result is rounded before storing.

LIST (*e* / **ATTRIBUTES OF** { *name* [**CALLED** *e*] / **EACH** *name* [**OF** *v*] }) *c*

A free-form output statement that labels and displays values of expressions and 1- and 2-dimensional arrays.

{ **LOOP** / **REPEAT** }

Used with DO to delimit a group of statements controlled by one or more logical control phrases.

MAIN

Marks the beginning of a program's main routine. Execution commences at the first executable statement after MAIN.

MOVE { **FROM** *e* / **TO** *v* }

Used within a routine defined for a monitored variable to access or set the value of the variable.

NEXT — See CYCLE

OTHERWISE — See ELSE

$$\text{NORMALLY [,]} \left\{ \begin{array}{l} \text{MODE} \left\{\begin{array}{l}\text{IS}\\ =\end{array}\right\} \left\{\begin{array}{l}\text{TEXT}\\ \text{INTEGER}\\ \text{REAL}\\ \text{ALPHA}\end{array}\right\} \\ \text{TYPE} \left\{\begin{array}{l}\text{IS}\\ =\end{array}\right\} \left\{\begin{array}{l}\text{SAVED}\\ \text{RECURSIVE}\end{array}\right\} \\ \left\{\begin{array}{l}\text{DIMENSION}\\ \text{DIM}\end{array}\right\} \left\{\begin{array}{l}\text{IS}\\ =\end{array}\right\} i \end{array} \right\}$$

Establishes background conditions for properties of variables and functions that are effective unless overridden by subsequent DEFINE declarations or, in the case of local arrays, first use.

NOW — See CALL

PERFORM — See CALL

$$\left[\begin{array}{l}\text{VERY OLD}\\ \text{OLD}\\ \text{NEW}\end{array}\right] \text{PREAMBLE}$$

Marks the beginning of the program preamble.

$$\text{PRINT } i \text{ [DOUBLE] LINES} \left[\text{WITH} \left\{\begin{array}{l} e \\ \text{A GROUP OF } \{e\}^{c} \text{ FIELDS}\end{array}\right\}^{c} \right]$$
$$\left[\text{SUPPRESSING FROM COLUMN } i\right] \left\{\begin{array}{l}\text{LIKE THIS}\\ \text{THUS}\\ \text{AS FOLLOWS}\end{array}\right\}$$

The i lines following the PRINT statement are format lines containing text and pictorial formats for the display of indicated expression values. The phrases

A GROUP OF $\{e\}^{c}$ FIELDS and SUPPRESSING FROM COLUMN i

can only be used within report sections that have column repetition.

PRIORITY ORDER IS $\{name\}^{c}$

Assigns a priority order to different classes of events.

```
PROCESS name [ ( THE | THIS | GIVEN | GIVING ) {name}^c
               ( {name}^c ) ]
```

```
PROCESSES [ INCLUDE {name}^c ]
```

REACTIVATE — See ACTIVATE

```
READ ( { v }^c [ AS ( BINARY | [(e)] {f2}^c ) ]
       AS {f1}^c ) USING ( THE BUFFER | [TAPE UNIT] e )
```

Used without an AS clause indicates a free-form data input.

```
RECORD [ MEMORY | STATUS ]
```

REGARDLESS — See ALWAYS

```
RELEASE {v}^c
```

Releases blocks of core pointed to by v; v's are assumed to be pointer variable

---

```
RELINQUISH e [UNITS OF] name [( e )]
```

```
REMOVE [THE] ( { FIRST | LAST } v
               [ THIS ABOVE ] e ) FROM [ THE THIS ] v
```

Removes an entity from a set.

REPEAT — See LOOP

REQUEST *e* **[UNITS OF]** *name* **[(** *e* **)]** **[, WITH PRIORITY** *e***]**

RESCHEDULE — See ACTIVATE

RESERVE $\left\{\{v\}^{C} \text{ AS } \{e\}^{\text{BY}} \text{ [BY *]}\right\}^{C}$

Allocates blocks of core of specified size to the pointer variables *v*. Words assigned are data if no BY* phrase appears, and are pointers otherwise.

RESET [THE] [*name*] **TOTALS OF** $\{v\}^{C}$

Initializes ACCUMULATE or TALLY counters associated with v. If TOTALS is not qualified by a word, all counters of v are initialized.

RESOURCES $\left[\text{INCLUDE } \{name\}^{C}\right]$

RESUME *name* **[CALLED** *v***]**

$\left\{\begin{matrix}\textbf{RESUME}\\ \textbf{SUPPRESS}\end{matrix}\right\}$ **SUBSTITUTION**

Used to override a currently defined substitution. This statement must not be placed on program cards with other statements.

RETURN $\left[\left\{\begin{matrix}(e)\\ \textbf{WITH } e\end{matrix}\right\}\right]$

Used as a procedure, a routine returns control to its calling program with the statement RETURN; used as a function, a routine returns control and a value to its calling program by either of the statements **RETURN(** *e* **)** or **RETURN WITH** *e*

REWIND [TAPE / UNIT] *e*

Rewinds an input/output device.

ROUTINE — See FUNCTION

SCHEDULE — See ACTIVATE

SKIP *e* { FIELDS / [INPUT / OUTPUT] { CARDS / LINES / RECORDS } }

Applies to the current input or current output unit. SKIP e FIELDS applies to the current input unit only when it is used for freeform data input. CARDS, LINES, and RECORDS are synonyms. If neither UNPUT nor OUTPUT is specified, INPUT is implied.

START NEW { PAGE / [INPUT / OUTPUT] { LINE / CARD / RECORD } }

Applies to the current input or current output unit. LINE, CARD, and RECORD are synonyms. If neither INPUT nor OUTPUT is specified, INPUT is implied.

START SIMULATION

Starts simulation by removing the first event from the events set and executing it.

STOP

Halts program execution.

STORE *e* IN *v*

Assigns a value to a variable without mode conversion.

SUBROUTINE — See FUNCTION

SUBSTITUTE $\left\{\begin{matrix}\text{THIS}\\\text{THESE}\end{matrix}\right\}$ *i* LINES FOR *w*

Similar to DEFINE TO MEAN but allows more than one card of words to be substituted.

SUBTRACT *e* FROM *v*

Subtracts the value of *e* from the value of the variable *v*.

SUPPRESS — See RESUME

SUSPEND [PROCESS]

The suspend statement may only appear in a process routine. The process is placed in the passive state.

[THE] SYSTEM

See EVERY

TALLY — See ACCUMULATE

$\left\{\begin{matrix}\text{TEMPORARY ENTITIES}\\\text{PERMANENT ENTITIES}\end{matrix}\right\}$ [INCLUDE $\{name\}^{c}$]

Declares the type of following EVERY statements to be permanent or temporary, as the case may be.

THE SYSTEM — See EVERY

[THEN] IF ϕ [,]

See IF

TRACE [USING $\left[\begin{matrix}\text{TAPE}\\\text{UNIT}\end{matrix}\right]$ *e*]

Produces a backtrack of current subprogram calls. When the SIMSCRIPT II operating system uses TRACE the standard output device (printer) is used.

UNLESS — See tc metaraviable (termination clause)

UNTIL — See sc metavariable (selection clause)

UPON — See EVENT

USE { THE BUFFER / [TAPE] [UNIT] e } FOR { INPUT / OUTPUT }

Sets the indicated input/output device as the current input or output unit. All subsequent input/output statements that do not specify their own devices in USING phrases use these current units. THE BUFFER causes reading and writing in an internal file.

{ WORK / WAIT } e { UNIT / DAYS / HOURS / MINUTES }

WHEN — See sc metavariable (selection clause)

WHILE — See tc metavariable (termination clause)

WITH — See sc metavariable (selection clause)

WORK — See WAIT

WRITE { { { e }^c AS { BINARY / [(e)] { f2 / s / * } } } / AS { f1 / s / * } } [USING { THE BUFFER / [TAPE] [UNIT] e }]

Attributes and Routines Required for Set Statements

SET STATEMENT	ATTRIBUTES REQUIRED						ROUTINES REQUIRED						
	F	L	P	S	M	N	FF	FL	FB	FA	RF	RL	RS
FILE FIFO	F	L		S				FL					
FILE LIFO	F			S			FF						
FILE ranked	F		P	S			FF						
FILE FIRST	F			S			FF						
FILE LAST	F	L		S				FL					
FILE BEFORE	F		P	S					FB				
FILE AFTER	F			S						FA			
REMOVE FIFO	F			S							RF		
REMOVE LIFO	F			S							RF		
REMOVE ranked	F			S							RF		
REMOVE FIRST	F			S							RF		
REMOVE LAST	F	L	P	S								RL	
REMOVE specific	F		P	S									RS
set IS EMPTY	F												
IS IN set					M								
FOR EACH...OF set	F			S									
FOR EACH...OF set ...REVERSE		L	P										
FOR EACH...FROM ...OF set				S									
FOR EACH...FROM ...OF set...REVERSE			P										
FOR EACH...AFTER ...OF set				S									
FOR EACH...AFTER ...OF set...REVERSE			P										
Automatic checking					M								

Statistical Keywords for ACCUMULATE Statement

STATISTICAL KEYWORD	SYNONYM	COMPUTATION	REQUIRED COUNTERS
NUMBER	NUM	N	N
SUM		$\Sigma X*(TIME.V-T_L)$	SUM, T_L
MEAN	AVG AVERAGE	$SUM/(TIME.V-T_0)$	SUM, T_L, T_0
SUM.OF.SQUARES	SSQ	$\Sigma X^2*(TIME.V-T_L)$	SSQ, T_L
MEAN.SQUARE	MSQ	$SSQ/(TIME.V-T_0)$	SSQ, T_L, T_0
VARIANCE	VAR	$MSQ - MEAN^2$	SSQ, SUM, T_L, T_0
STD.DEV	STD	SQRT.F(VAR)	SSQ, SUM, T_L, T_0
MAXIMUM	MAX	M = maximum (X) for all X	M, N
MINIMUM	MIN	m = minimum (X) for all X	m, N

NOTES:

TIME.V	current simulated time
T_L	simulated time at which variable was set to its current value
T_0	simulated time at which accumulation started
X	sample value of accumulation variable (before it changes to a new value)

Statistical Keywords for TALLY Statement

STATISTICAL KEYWORD	SYNONYM	COMPUTATION	REQUIRED COUNTERS
NUMBER	NUM	N	N
SUM		ΣX	SUM
MEAN	AVG AVERAGE	SUM/N	SUM, N
SUM.OF.SQUARES	SSQ	ΣX^2	SSQ
MEAN.SQUARE	MSQ	SSQ/N	SSQ, N
VARIANCE	VAR	MSQ - MEAN^2	SSQ, SUM, N
STD.DEV	STD	SQRT.F(VAR)	SSQ, SUM, N
MAXIMUM	MAX	M = maximum (X) for all X	M, N
MINIMUM	MIN	m = minimum (X) for all X	m, N

NOTE: X is the sample value of tallied variable (before it changes to a new value).

Statement Type	Statement	Rules
1a 1b 1c 1d 1e 1f	NORMALLY DEFINE TO MEAN SUBSTITUTE SUPPRESS SUBST. RESUME SUBST. GENERATE INHIBIT	Can appear anywhere in preamble.
2a 2b 2c	TEMPORARY ENTITIES PERMANENT ENTITIES EVENT NOTICES	A preamble may contain many Type 2a, 2b, and 2c statements. Each may be followed by a group of Type 3a, 4, and 5 statements.
3a 3b	EVERY THE SYSTEM	Many can follow a Type 2 statement. An entity or event notice name can appear in more than one EVERY statement.
4	DEFINE VARIABLE	No precedence relation if it defines a global variable. Must follow all Type 3a statements if it defines an attribute named in them. A variable, attribute, or function name can appear in only one DEFINE statement.
5	DEFINE SET	Must follow Type 4 statements in a Type 2 statement group if it qualifies a set named in them.
6a 6b 6c	BREAK TIES EXTERNAL EVENTS EXTERNAL UNITS	One statement allowed for each event notice.
7	PRIORITY	Must follow all Type 2c and 6b statements.
8a 8b	BEFORE AFTER	Allowed for each temporary entity, set, and event notice.
9a 9b	TALLY ACCUMULATE	Either TALLY or ACCUMULATE statistics but not both, may be specified for unsubscripted global variables, attributes of permanent or temporary entities, event notices, processes or resources.

Of these statements, only Types 1 and 4 can be used in routines to declare local background conditions, variables, and substitutions.

SYSTEM DEFINED NAMES

SYSTEM DEFINED VALUES

Variable	Mode	Description	Default
BETWEEN.V	Subprogram	Subprogram variable called before each event is executed.	0
BUFFER.V	Integer	The length of the internal buffer.	132
EOF.V	Integer	End-of-file code; zero denotes notes that an end-of-file marker is an error; one indicates return control with EOF.V set to 2 when end-of-file is encountered; one for each input unit.††	0
EVENT.V	Integer	Code representing the event class to occur next.	0
EVENTS.V	Integer	The number of event classes.	0
F.EV.S(*)	Integer	Array containing the first-in-set pointers for the event set, EV.S	0
HEADING.V	Integer	A subprogram variable tested by the system for each new page.††	0
HOURS.V	Real	Number of hours per simulated day.	24
LINE.V	Integer	Number of the current output line.††	1
LINES.V	Integer	Number of lines per page.††	55
L.EV.S (*)	Integer	Array containing the last-in-set pointers for the event set EV.S	0
MARK.V	Alpha	Termination character required on external event cards and on the input for random variables.	*
MINUTES.V	Real	Number of minutes per simulated hour.	60

†† A separate value is maintained for each unit; only the currently used value is accessible to the program.

Variable	Mode	Description	Default
N.EV.S (n)	Integer	Function returning the number of pending event notices of class n.	
PAGE.V	Integer	Number of the current page.††	1
PAGECOL.V	Integer	If ≠ 0, column number in which the word PAGE and the value of PAGE.V is to be printed on the output listing.††	0
PARM.V (*,*)	Alpha	Array containing the user parameters passed to the program.	
PRMB.V (*)	Integer	Array containing the values of the attributes of THE SYSTEM with IN WORD clauses.	
RCOLUMN.V	Integer	Pointer to the last column read in the input buffer.††	0
READ.V	Integer	Number of the current input unit.	5
RECORD.V(n)	Integer	The number of records read from or written on the indicated unit.	0
RRECORD.V	Integer	The number of records read from the current input unit.††	0
SEED.V	Integer	Array containing initial random numbers.	Yes
TIME.V	Real	Current simulated time.	0
WCOLUMN.V	Integer	Pointer to the column last written in the output buffer.††	0
WRECORD.V	Integer	The number of records written on the current output unit.††	0
WRITE.V	Integer	Number of the current output unit.	6

††A separate value is maintained for each unit; only the currently used value is accessible to the program.

SYSTEM DEFINED CONSTANTS		
Constant	Mode	Description
EXP.C	Real	2.7182818284590452
INF.C	Integer	Largest INTEGER value that can be stored.
PI.C	Real	3.1415926535897932
RADIAN.C	Real	57.295779513082321 degrees/ radian
RINF.C	Real	Largest REAL value that can be stored.

SYSTEM DEFINED FUNCTIONS

Function Mnemonic	Arguments†	Function Mode	Description
ABS.F	e	Mode of e	Returns the absolute value of the expression.
ALPHA.F	a,b	Integer	Compares two alphanumeric words a and b and returns: —1 if a<b 0 if a=b 1 if a>b
AND.F	a,b	Integer	Logical product of a and b.
ARCCOS.F	e	Real	Computes the arc cosine of a real expression; $-1 \geqslant e \geqslant 1$
ARCTAN.F	e_1, e_2	Real	Computes the arc tangent of e_1/e_2; $(e_1, e_2) \neq (0.0)$
BETA.F	e_1, e_2, e_3	Real	Returns a random sample from a beta distribution e_1 = power of x, real; $e_1 \geqslant 0$ e_2 = power of (1−x), real; $e_2 \geqslant 0$ e_3 = random number stream, integer
BINOMIAL.F	e_1, e_2, e_3	Integer	Returns a random sample from a binomial distribution e_1 = number of trials, integer e_2 = probability of success, real e_3 = random number stream, integer
COS.F	e	Real	Computes the cosine of a real expression given in radians.

†e = expression that can be of any complexity, including functions
v = variable

DATE.F	e_1, e_2, e_3	Real	Converts a calendar date to cumulative simulation time, based on values given to ORIGIN.R e_1 = month, integer e_2 = day, integer e_3 = year, integer
DAY.F	e	Integer	Converts simulation time to the day portion based on values given to ORIGIN.R e = cumulative simulation time, real
DIM.F	v(*)	Integer	Returns the number of elements pointed to by the pointer variable v, in the dimension of the array v.
DIV.F	e_1, e_2	Integer	Returns the truncated value of (e_1/e_2) e_1 = dividend, integer e_2 = divisor, integer; $e_2 \neq 0$
EFIELD.F	none	Integer	Returns the ending column of the next data field to be read by a READ Free Form statement.
ERLANG.F	e_1, e_2, e_3	Real	Returns a sample value from an Erlang distribution e_1 = mean, real e_2 = k, integer e_3 = random number stream, integer

Function Mnemonic	Arguments	Function Mode	Description
EXP.F	e	Real	Computes EXP.C to the e^{th} power; e must be real.
EXPONEN-TIAL.F	e_1, e_2	Real	Returns a random sample from an exponential distribution e_1 = mean, real e_2 = random number stream, integer
FRAC.F	e	Real	Returns the fractional portion of a real expression.
GAMMA.F	e_1, e_2, e_3	Real	Returns a random sample from a gamma distribution e_1 = mean, real e_2 = k, real e_3 = random number stream, integer
GAMMAJ.F	e_1, e_2, e_3	Real	Same as GAMMA.F
HOUR.F	e	Integer	Converts event time to the hour portion e = cumulative event time, real
INT.F	v,e	Integer	Returns the rounded integer portion of a real expression.
ISTEP.F	v, e	Integer	Returns a random sample from a look-up table without interpolation v = variable that points to the look-up table e = random number stream, integer

Function Mnemonic	Arguments	Function Mode	Description
ITOA.F	e	Alpha	Converts an integer expression to an alpha-numeric value, left adjusted in a blank field.
LIN.F	v, e	Real	Returns a random sample from a look-up table, using linear interpolation v = variable that points to the look-up table e = random number stream integer
LOG.E.F	e	Real	Computes the natural logarithm of a real expression e > 0.
LOG. NORMAL.F	e_1, e_2, e_3	Real	Returns a random sample from a lognormal distribution e_1 = mean, real e_2 = standard deviation, e_3 = random number stream, integer
LOG.10.F	e	Real	Computes $\log_{10}$ of a real expression; e > 0.
MASK.F	e	Integer	Builds a left justified mask of e bits.

Function Mnemonic	Arguments	Function Mode	Description
MAX.F	$e_1, e_2, \ldots, e_n$	Real if any e_i real; if none, integer	Returns the value of the largest e_i.
MIN.F	$e_1, e_2, \ldots, e_n$	Real if any e_i real; if none, integer	Returns the value of the smallest e_i.
MINUTE.F	e	Integer	Converts event time to the minute portion e = cumulative event time, real
MOD.F	e_1, e_2	Real if either e_i real; if none, integer	Computes a remainder as e_1-TRUNC.F(e_1/e_2) *e_2; $e_2 \neq 0$
MONTH.F	e	Integer	Converts simulation time to month portion based on values given to ORIGIN.R e = cumulative simulation time, real
NDAY.F	e	Integer	Converts event time to the day portion e = cumulative event time, real
NORMAL.F	e_1, e_2, e_3	Real	Returns a random sample from a normal distribution $e_1 \equiv$ mean, real e_2 = standard deviation, real e_3 = random number stream, integer
OR.F	a, b	Integer	Logical sum of a and b.

Function Mnemonic	Arguments	Function Mode	Description
OUT.F	e	Alpha	Sets or returns the alphanumeric value of the e^{th} character in the current output buffer; e must yield an integer value; $e \geq 0$; both right- and left-handed function.
POISSON.F	e_1, e_2	Integer	Returns a random sample from a Poisson distribution e_1 = mean, real e_2 = random number stream, integer
RANDI.F	e_1, e_2, e_3	Integer	Returns a random sample uniformly distributed between a range of values e_1 = beginning value, integer e_2 = ending value, integer e_3 = random number stream, integer
RANDOM.F	e	Real	Returns a pseudo-random number between zero and one e = random number stream, integer
REAL.F	e	Real	Converts an integer expression to a real value.

Function Mnemonic	Argument	Function Mode	Description
RSTEP.F	v, e	Real	Returns a random sample from a look-up table v = variable that points to the look-up table e = random number stream, integer
SFIELD.F	none	Integer	Returns the starting column of the next data field to be read by a READ Free Form statement.
SHL.F	w, n	Integer	Shift w left n positions, end around.
SHR.F	w, n	Integer	Shift w right n positions, arithmetic.
SIGN.F	e	Integer	Indicates the sign of a real expression. 1 if e > 0 0 if e = 0 -1 if e < 0
SIN.F	e	Real	Computes the sine of a real expression given in radians.
SQRT.F	e	Real	Computes the square root of a real expression; e ⩾ 0.
TAN.F	e	Real	Computes the tangent of a real expression given in radians.
TRUNC.F	e	Integer	Returns the truncated integer value of a real expression.

Function Mnemonic	Arguments	Function Mode	Description
UNIFORM.F	e_1, e_2, e_3	Real	Returns a uniformly distributed random sample between a range of values e_1 = beginning value, real e_2 = ending value, real e_3 = random number stream, integer
WEEKDAY.F	e	Integer	Converts event time to the weekday portion e = cumulative event time, real
WEIBULL.F	e_1, e_2, e_3	Real	Returns a sample value from a Weibull distribution e_1 = scale parameter, real e_2 = shape parameter, real e_3 = random number stream, integer
XOR.F	a, b	Integer	Logical difference of a and b.
YEAR.F	e	Integer	Converts simulation time to the year portion based on values given to ORIGIN.R e = cumulative simulation time, real
ZTIME.F	none	Real	Returns the elapsed execution time in seconds.

SYSTEM DEFINED ROUTINES		
Routine	Arguments	Description
ORIGIN.R	e_1, e_2, e_3	Establishes an origin time when the calendar format is used e_1 = month, integer e_2 = day, integer e_3 = year, integer
TIME.R	none	Controls simulation timing and selects events.
SNAP.R	none	User supplied routine called by SIMSCRIPT II.5 when an execution error is detected.
DATE.R	y_1, y_2	Yields the current date and current time of day from the operating system. y_1 = date, alpha y_2 = time, alpha
EOF.R	e	Writes an end of file on logical unit e.
EXIT.R	e	Stops the simulation after printing EXIT e and issuing a traceback.
MOVE.F	v_1, v_2, n	Move n words from one storage area to another. v_1 = Base pointer of source array v_2 = Base pointer of target array n = Number of words

GENERATED VARIABLES, ATTRIBUTES AND ROUTINES			
Generated for	Generated Elements	Name	Definition
Accumulated and tallied variables	Routine	Q.variable	A left-hand monitoring routine that accumulates or tallies data
Entities (5 unique characters)	Variables	Gentity*	Global variable having the entity class name
		N.entity	No. of entities of the entity class (permanent entities only)
		K.entity	Size of entity in computer words (temporary entities only)
	Routines	C.entity	To reserve storage for permanent entities (i.e., to create them)
		D.entity	Called when destroying a temporary entity to check for set membership error
		A.entity	Called to list the values of entity attributes
Event Notices (5 unique characters)	Variables	Gevent*	Global variable having the event notice name
		I.event	Global variable holding the subscript for this event class in the event set

*The G prefix entry name is generated only when a word or array number is not specified for the indicated name. If this entry is not generated, and associated routine and variable names are not present, there is no restriction on the length of the name.

Generated for	Generated Elements	Name	Definition
		K.event	Size of the event notice in computer words
	Routines	C.event	Files events, whose priorities are declared in BREAK TIES statements, in the proper event set
		D.event	Called when destroying event notice to check for set membership error
		A.event	Called to list values of event notice attributes
Event notice Records	Attributes	TIME.A	Time event is to occur
		EUNIT.A	Equals 0 for an endogenous event; equals input unit number ($\neq 0$) for an exogenous event
		P.EV.S	Pointer to predecessor event in the event set
		S.EV.S	Pointer to successor event in the event set
		M.EV.S	Set to 1 if the event is in the set; set to 0 if the event is not in the set
Random Variables	Attributes of RANDOM.E	PROB.A	Probability value
		IVALUE.A RVALUE.A	Sample value: IVALUE.A contains an integer value; RVALUE.A contains a real value
		S.variable	Pointer to successor

Generated for	Generated Elements	Name	Definition
Sets (5 unique characters)	Attributes of owner entities	F.set	Pointer to first entity in set
		L.set	Pointer to last entity in set
		N.set	No. of entities currently in the set
	Attributes of member entities	P.set	Pointer to predecessor entity in set
		S.set	Pointer to successor in set
		M.set	Equals 1 if the entity is in the set; equals 0 if the entity is not in the set
	Routines	T.set	Files entity first or ranked in set
		U.set	Files entity last in set
		V.set	Files entity before specified entity in set
		W.set	Files entity after specified entity in set
		X.set	Removes first entity from set
		Y.set	Removes last entity from set
		Z.set	Removes specific entity from set

SIMII5 COMPILATION CONTROL CARD

{ SIMII5.
SIMII5, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8. }

All the p_i parameters are order independent.

– p_1	Source input	. absent	INPUT assumed
		. I = 1fn	Source input on file 1fn
– p_2	Binary output	. absent	LGO assumed
		. B = 1fn	Relocatable binary on file 1fn
		. B = 0	No binary output
– p_3	Source listing	. absent	Source listing on file OUTPUT
		. L = OUTPUT	
		. L = 0	Suppresses listing output except for errors
		. L = 1fn	Source listing on file 1fn
– p_4	Star, allstar and object listing	. absent	No listing
		. S = 1fn	Star listing on file 1fn
		. O = 1fn	Object listing on file 1fn
		. A = 1 fn	Allstar listing on file 1 fn
		. OS = 1fn	Star and object listing on file 1 fn
		. AO = 1 fn	Allstar and object listing on file 1 fn
– p_5	Compilation options	. OPT = $f_1 f_2 \ldots f_n$	
		. fi = T	Generate line numbers for traceback
		. fi = P	Continue compilation even if errors are detected in the PREAMBLE
		. fi = E	Do not abort the job if compilation errors are detected
		. fi = 8	Print output at 8 lines/inch
		. fi = 6	Print output at 6 lines/inch
		. fi = N	Suppress listing of the PREAMBLE
		. fi = C	Compress source listing
		. fi = V	Use small buffers for compilation
– p_6	Last column	. LC = n	Last card column to be scanned by the compiler
– p_7	Symbolic reference map	. absent	R = 1
		. R = 1	Local map, no line numbers
		. R = 2	Global map, no line numbers
		. R = 3	Local and global map, no line numbers
		. R = 4	Global map with line numbers
		. R = 5	Local map, no line numbers Global map with line numbers

		. R = 6	Local map with line numbers
		. R = 7	Local map with line numbers Global map, no line numbers
		. R = 8	Local and global map with line numbers
– p_8	. Structured IF	. absent	IF = NEW (i.e. new IF-ELSE-ALWAYS construct)
		. IF = OLD	Unstructured IF

Examples:

SIMII5.

- Source input on INPUT
- Source listing on OUTPUT
- Relocatable binary on LGO
- No star or assembly listings
- No line numbers
- Abort job if errors in the PREAMBLE or compilation errors
- Last column = 80
- IF = NEW and R = 1

SIMII5, I = COMPILE, LS = OUTPUT, OPT = PTE8, LC = 72.

- Source input on file COMPILE
- Source listing and star listing on file OUTPUT (Note: If all the listings are directed to the same file, the options can be merged)
- Continue compilation even if errors are detected in the PREAMBLE
- Include line numbers for traceback
- Execute the program even if compilation errors occurred
- Print the listings at 8 lines/inch
- Last column = 72.

SIMII5 EXECUTION CONTROL CARD

{ LGO.
LGO, $f_1, f_2, \ldots f_n, p_1, p_2, p_3$. }

. $f_1, f_2, \ldots f_n$ — External file name substitution
SIMUn = 1fn
n = internal logical unit number $0 \leqslant n \leqslant 9$
– even numbered units are carriage control files
– odd numbered units are non-slew BCD files
– any unit number can be used for binary files. (except 5 and 6)
1fn = external file name
FORTRAN file name definition
TAPEn = 1fn
FTN routines must be compiled with SYSEDIT = FILES
All files used by FORTRAN must be declared.
FORTRAN cannot share SIMSCRIPT files.

. p_1 – Print limit — PL = n — n is the maximum number of decimal lines to print on the OUTPUT file (default: 10000)

. p_2 – Dynamic storage — DS = n — n is the octal or decimal number of words reserved initially for dynamic storage. Assumed octal. (Ex: 4000 or 4000B, 10000D)

. p_3 – User parameters — PARM = $g_1, g_2, \ldots g_n$; g_i represents either a single value or a group of values separated by an =, a + or a — sign.

Examples:

LGO.
– SIMU5 = INPUT
– SIMU6 = OUTPUT
– No FORTRAN files
– Print limit on output = 10000

LGO, SIMU5 = INPT, SIMU8 = REPORT, PL = 15000, DS = 20000B.
– Input on file INPT
– SIMU6 = OUTPUT
– Additional file SIMU8 = REPORT
– Print limit on OUTPUT = 15000 lines
– Initial dynamic storage = 20000B

CDC Graphic	ASCII Graphic Subset	Display Code	Hollerith Punch (026)	External BCD Code	ASCII Punch (029)	ASCII Code	CDC Graphic	ASCII Graphic Subset	Display Code	Hollerith Punch (026)	External BCD Code	ASCII Punch (029)	ASCII Code
:↑	:	00†	8-2	00	8-2	3A	6	6	41	6	06	6	36
A	A	01	12-1	61	12-1	41	7	7	42	7	07	7	37
B	B	02	12-2	62	12-2	42	8	8	43	8	10	8	38
C	C	03	12-3	63	12-3	43	9	9	44	9	11	9	39
D	D	04	12-4	64	12-4	44	+	+	45	12	60	12-8-6	2B
E	E	05	12-5	65	12-5	45	−	−	46	11	40	11	2D
F	F	06	12-6	66	12-6	46	*	*	47	11-8-4	54	11-8-4	2A
G	G	07	12-7	67	12-7	47	/	/	50	0-1	21	0-1	2F
H	H	10	12-8	70	12-8	48	(	(	51	0-8-4	34	12-8-5	28
I	I	11	12-9	71	12-9	49	)	)	52	12-8-4	74	11-8-5	29
J	J	12	11-1	41	11-1	4A	$	$	53	11-8-3	53	11-8-3	24
K	K	13	11-2	42	11-2	4B	=	=	54	8-3	13	8-6	3D
L	L	14	11-3	43	11-3	4C	blank	blank	55	no punch	20	no punch	20
M	M	15	11-4	44	11-4	4D	, (comma)	, (comma)	56	0-8-3	33	0-8-3	2C
N	N	16	11-5	45	11-5	4E	. (period)	(period)	57	12-8-3	73	12-8-3	2E
O	O	17	11-6	46	11-6	4F	≡	#	60	0-8-6	36	8-3	23
P	P	20	11-7	47	11-7	50	[	[	61	8-7	17	12-8-2	5B
Q	Q	21	11-8	50	11-8	51	]	]	62	0-8-2	32	11-8-2	5D
R	R	22	11-9	51	11-9	52	%	%	63††	8-6	16	0-8-4	25
S	S	23	0-2	22	0-2	53	≠	" (quote)	64	8-4	14	8-7	22
T	T	24	0-3	23	0-3	54	→	(underline)	65	0-8-5	35	0-8-5	5F
U	U	25	0-4	24	0-4	55	∨	_	66	11-0 or 11-8-211††	52	12-8-7 or 11-0††	21
V	V	26	0-5	25	0-5	56							
W	W	27	0-6	26	0-6	57	∧	&	67	0-8-7	37	12	26
X	X	30	0-7	27	0-7	58	↑	' (apostrophe)	70	11-8-5	55	8-5	27
Y	Y	31	0-8	30	0-8	59	↓	?	71	11-8-6	56	0-8-7	3F
Z	Z	32	0-9	31	0-9	5A	<	<	72	12-0 or 12-8-211††	72	12-8-4 or 12-0††	3C
0	0	33	0	12	0	30							
1	1	34	1	01	1	31	≥	>	73	11-8-7	57	0-8-6	3E
2	2	35	2	02	2	32	∨	@	74	8-5	15	8-4	40
3	3	36	3	03	3	33	∧	\	75	12-8-5	75	0-8-2	5C
4	4	37	4	04	4	34	⌐	^ (circumflex)	76	12-8-6	76	11-8-7	5E
5	5	40	5	05	5	35	; (semicolon)	; (semicolon)	77	12-8-7	77	11-8-6	3B

† Twelve or more zero bits at the end of a 60-bit word are interpreted as end-of-line mark rather than two colons. End-of-line mark is converted to external BCD 1632.

†† In installations using the CDC 63-graphic set, display code 00 has no associated graphic or Hollerith code; display code 63 is the colon (8-2 punch). The % graphic does not exist and translations from ASCII/EBCDIC % yield a blank (55_8).

††† The alternate Hollerith (026) and ASCII (029) punches are accepted for input only.

CORPORATE HEADQUARTERS
8100 34TH AVENUE SOUTH
MINNEAPOLIS, MINNESOTA
MAILING ADDRESS • BOX 0, MPLS., MINN. 55440

SALES OFFICES AND SERVICE CENTERS
IN MAJOR CITIES THROUGHOUT THE WORLD