AD-A058 724 PRITSKER AND ASSOCIATES INC WEST LAFAYETTE IND F/G 9/2
THE SAINT USER'S MANUAL.(U)
JUN 78 D B WORTMAN, S D DUKET, D J SEIFERT F33615-76-C-5012
UNCLASSIFIED AMRL-TR-77-62 NL

1 OF 2
AD A058724



AD A058724

DDC FILE COPY

AMRL-TR-77-62

LEVEL II

THE SAINT USER'S MANUAL

DAVID B. WORTMAN
STEVEN D. DUKET
PRITSKER & ASSOCIATES, INC.
P. O. BOX 2413
WEST LAFAYETTE, INDIANA 47906

DEBORAH J. SEIFERT
REUBEN L. HANN
GERALD P. CHUBB
AEROSPACE MEDICAL RESEARCH LABORATORY

DDC
SEP 14 1978

JUNE 1978

Approved for public release; distribution unlimited.

AEROSPACE MEDICAL RESEARCH LABORATORY
AEROSPACE MEDICAL DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

78 09 12 025

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with Defense Documentation Center should direct requests for copies of this report to:

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314

TECHNICAL REVIEW AND APPROVAL

AMRL-TR-77-62

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

CHARLES BATES, JR.
Chief
Human Engineering Division
Aerospace Medical Research Laboratory

AIR FORCE/56780/31 July 1978 — 500

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AMRL-TR-77-62	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): THE SAINT USER'S MANUAL	5. TYPE OF REPORT & PERIOD COVERED: Final Report, Aug 75-Dec 76 / 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): David B. Wortman*, Steven D. Duket*, Deborah J. Seifert**, Reuben L. Hann**, Gerald P. Chubb**	8. CONTRACT OR GRANT NUMBER(s): F33615-76-C-5012
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Pritsker & Associates*, P.O. Box 2413, West Lafayette, IN 47906; Aerospace Med. Research Lab.**, WPAFB, OH 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62202F; 7184-13-03
11. CONTROLLING OFFICE NAME AND ADDRESS: Aerospace Medical Research Laboratory, Aerospace Medical Division, AFSC, Wright-Patterson Air Force Base, OH 45433	12. REPORT DATE: Jun 78 / 13. NUMBER OF PAGES: 142
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computers; Modeling; Operator Loading; Operations Research; Mission Analysis; Survivability/Vulnerability; Man-Machine Systems; Crew Performance; Networks; Simulation; Discrete Event Simulation; Next Event Simulation; Continuous Simulation; Combined Simulation; Simulation Languages; SAINT

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The procedures for using the SAINT simulation program to analyze system models are described in detail. SAINT (Systems Analysis of Integrated Networks of Tasks) is a network modeling and simulation technique developed to assist in the design and analysis of complex man-machine systems. SAINT consists of a symbol set for modeling systems and a computer program for analyzing such models. SAINT provides the conceptual framework

over

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391 382

for representing systems that consist of discrete task elements, continuous state variables, and interactions between them. While SAINT was designed for modeling manned systems in which human performance is a major concern, it is potentially applicable to a broad class of systems--those in which discrete and continuous elements are to be portrayed and quantified and whose behavior exhibits time-varying properties. SAINT provides a mechanism for describing these dynamics so a systematic assessment can be made of the relative contribution system components make to overall system performance. This report is intended as a procedural manual and reference guide for those persons already having some knowledge of the basic SAINT concepts and symbol set. Some of the topics covered are: (1) preparation of input data, (2) a description of the functions performed by the user-written sub-programs included in SAINT, (3) the procedures for using the available user support programs, (4) a listing of the errors which SAINT can detect along with their probable causes, and (5) information concerning the operation and structure of the SAINT program itself.

SUMMARY

This report describes the procedures for using the SAINT simulation program to analyze system models. SAINT (Systems Analysis of Integrated Networks of Tasks) is a network modeling and simulation technique designed to aid the system designer and human engineer in analyzing complex man-machine systems. It provides the conceptual framework which allows the development of system models in which men, machines, and the environment are represented. It permits the assessment of the effect of the component characteristics of the system on overall system performance. The symbolism and terminology required for modeling systems using SAINT are introduced and described in Simulation Using SAINT: A User-Oriented Instruction Manual (1). The overall structure and individual FORTRAN subprograms of SAINT are described in Documentation for the SAINT Simulation Program (2). The use of an external statistical analysis package to analyze SAINT output is described in Analyzing SAINT Output Using SPSS (3).

In this manual, the SAINT symbol set and terminology that are presented in detail in Simulation Using SAINT: A User-Oriented Instruction Manual (1) are summarized. The preparation of input data and the organization of the input deck are described. The functions performed by the user-written subprograms included in SAINT are defined. The procedures for using the available user support subprograms are presented. The errors that SAINT can detect, as well as the probable cause of each error, are provided. Information concerning the operation and structure of the SAINT simulation program is also presented.

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
SPECIAL
A

PREFACE

This study was initiated by the Human Engineering Division, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio 45433. The research was conducted by Pritsker & Associates, Inc., P. O. Box 2413, West Lafayette, Indiana 47906. The work was performed in support of Project 7184 "Man-Machine Integration Technology," task 718413 "Man-Machine Models for System Performance Assessment." The research sponsored by this contract was performed between August, 1975 and December, 1976, under Air Force contract F33615-76-C-5012.

In developing the SAINT technique, the authors received guidance and assistance from many individuals, especially the staff members of Pritsker & Associates and the Aerospace Medical Research Laboratory. We appreciate their efforts and acknowledge their contributions.

TABLE OF CONTENTS

	Page
SECTION I - INTRODUCTION	9
SECTION II - SAINT SYMBOLISM	10
Task Input	10
Task Description	12
ATAS	12
DIFF	14
DMOD	14
INCM	15
LABL	15
MARK	15
MODF	16
PREC	16
PRTY	17
RCLR	17
REGL	17
RESR	18
STAT	18
SWIT	19
TCLR	19
TIME	20
UTCH	20
Examples of Task Descriptions	20
Example 1	21
Example 2	21
Example 3	21
Example 4	22
Task Output	22
Available Branching Types	23
Probabilistic Branching Codes	24
Conditional Branching Codes	25
Additional Task Modeling Concepts	27
State Variable Definitions	27
Monitors	27
LABL	30
MONF	30
MTAS	30
MSWT	30
Example of Monitor Description	31

Page

SECTION III - SAINT DATA INPUT PROCEDURE..... 32

Data Input Card Preparation..... 32

Detailed Description of Data Types..... 38
GEN - General Information..... 39
SGE - State Variable General Information..... 40
POP - Program Options..... 41
OUT - Output Options..... 42
DIS - Distribution Set Definitions..... 44
Available SAINT Distributions..... 45
Parameters Required for SAINT Distributions.... 45
UBO - User-Generated Statistics for Variables Based on Observations..... 47
UTI - User-Generated Statistics for Time Persistent Variables..... 48
UHI - User-Generated Histograms..... 49
User-Generated Plots/Tables..... 50
UPL - User-Generated Plot/Table Characteristics..... 51
UVA - User-Generated Plot/Table Variable Information..... 52
IMO - Initial Moderator Function Status Specification..... 54
IRA - Initial Resource Attribute Specification. 55
ISA - Initial System Attribute Specification... 56
LRE - Labels for Resources..... 57
LSV - Labels for State Variables..... 58
Task Definitions..... 59
TAS - Task Characteristics..... 60
STA - Statistics Task Information..... 62
UTC - Specification of User-Defined Task Characteristics..... 63
MOD - Specification of Moderator Function Status..... 64
ATA - Specification of Attribute Assignment Information..... 65
Successor Tasks..... 66
DET - Deterministic Branching..... 67
PRO - Probabilistic Branching..... 68
CFI - Conditional - Take First Branching..... 69
CAL - Conditional - Take All Branching..... 71
NMO - Network Modifications..... 72
DMO - Distribution Set Modifications..... 73
TCL - Task Clearing..... 74
RCL - Resource Clearing..... 75
SWI - Switching Information..... 76
REG - Regulation Information..... 77
State Variable Monitor Information..... 78

Page

MON - Monitor Characteristics..... 79
MTA - Monitor Actuated Task Signaling..... 80
MSW - Monitor Actuated Switching..... 81
SST - State Variable Statistics Information.... 82
State Variable Plots/Tables..... 83
PLO - State Variable Plot/Table Characteristics..... 83
VAR - State Variable Plot/Table Variable Information..... 84

Condensed Data Input Description..... 86

SECTION IV - SAINT USER-WRITTEN SUBPROGRAMS..... 91

Subroutine ENDIT..... 92

Subroutine INTLC..... 92

Subroutine MODRF..... 92

Function PRIOR..... 92

Subroutine STATE..... 92

Subroutine UERR..... 93

Subroutine UINPT..... 93

Function USERF..... 93

Subroutine UOTPT..... 93

SECTION V - SAINT USER SUPPORT SUBPROGRAMS..... 94

Attribute Values..... 95
Subroutine GETIA..... 95
Subroutine PUTIA..... 95
Subroutine GETRA..... 95
Subroutine PUTRA..... 95
Subroutine GETSA..... 95
Subroutine PUTSA..... 95

User-Defined Task Characteristics..... 96
Subroutine GETTC..... 96
Subroutine PUTTC..... 96

AD-A109 035

Page

Task Scheduling..... 97
Subroutine GETPR..... 97
Subroutine PUTPR..... 97
Subroutine QRANK..... 97
Function TIMEQ..... 97
Function TMARK..... 97

Random Deviate Generators..... 98
Function BETA..... 98
Function DRAND..... 98
Function ERLNG..... 98
Function GAMM..... 98
Subroutine NPOSN..... 98
Function RLOGN..... 98
Function RNORM..... 98
Function TRNGL..... 98
Function UNFRM..... 98
Function WEIBL..... 98

User-Generated Statistics..... 99
Subroutine CLEAR..... 99
Subroutine CLRHI..... 99
Subroutine CLROB..... 99
Subroutine CLRPT..... 99
Subroutine CLRTP..... 100
Subroutine UCLCT..... 100
Subroutine UHIST..... 100
Subroutine UPLOT..... 100
Subroutine UTMSA..... 101
Subroutine UTMST..... 101

Simulation Status..... 102
Subroutine DMPAR..... 102
Subroutine DMPMF..... 102
Subroutine DMPRS..... 102
Subroutine DMPSA..... 102
Subroutine DMPST..... 102
Subroutine DMPSV..... 102

SECTION VI - SAINT INPUT ERRORS..... 102

SECTION VII - SAINT EXECUTION ERRORS..... 104

SECTION VIII - SAINT OPERATING PROCEDURES..... 119

SECTION IX - SAINT OVERLAY STRUCTURE..... 120

Page

SECTION X - DIMENSION REQUIREMENTS FOR SAINT ARRAYS.... 122

REFERENCES..... 134

BIBLIOGRAPHY..... 135

LIST OF ILLUSTRATIONS

Figure

1 Task Symbol..... 11

2 Task Modification..... 28

3 Task Signaling as a Result of Task or Resource Clearing..... 28

4 Task Signaling as a Result of Threshold Crossing..... 28

5 State Variable Information on Network Model. 29

6 State Variable Monitor Symbol..... 29

7 Main Program for SAINT..... 127

LIST OF TABLES

Table

1 Task Description Codes..... 13

2 Available Branching Types..... 23

3 Probabilistic Branching Codes..... 24

4 Conditional Branching Codes..... 25

5 List of SAINT Input Data Types..... 33

6 Condensed SAINT Data Input Description..... 87

7 SAINT User-Written Subprograms..... 92

Table		Page
8	SAINT User Support Subprograms: Attribute Values	95
9	SAINT User Support Subprograms: General Task Characteristics	96
10	SAINT User Support Subprograms: Task Scheduling	97
11	SAINT User Support Subprograms: Random Deviate Generators	98
12	SAINT User Support Subprograms: User-Generated Statistics	99
13	SAINT User Support Subprograms: Simulation Status	102
14	SAINT Execution Error Descriptions	105
15	Groupings of SAINT Subprograms	121
16	Dimension Requirements for SAINT Arrays	123
17	SAINT COMMON Block Characteristics	129
18	COMMON Blocks Included in SAINT Subprograms	130

SECTION I

INTRODUCTION

SAINT provides a unique and powerful capability for modeling and analyzing complex man-machine systems. The technique's conceptual framework allows the development of system models in which men, machines, and the environment are represented. This permits an analyst to investigate the impact of modifications to the man-machine-environment interface on overall system performance. In addition, such investigations can be performed without a major investment in equipment and time and without necessitating a commitment to prototype hardware development.

SAINT enables an analyst to input a description of the system to be analyzed. The system description includes the tasks performed by the resources, the precedence relations among tasks, the flow of information through the system, and the effects of environmental stressors on task performance. SAINT also allows the specification, evaluation, and monitoring of state variables which represent processes that change status continously over time. In addition, modeling capabilities are available for representing the dynamic interaction of tasks, resources, and state variables in an overall systems context.

The system description serves as input data to the SAINT simulation program. SAINT automatically performs an analysis of the model developed and provides summary information concerning resource utilization, task performance, state variable status, and a wide variety of other system performance measures. In addition, SAINT allows the user to develop his own special summary reports.

The SAINT simulation technique is documented in three manuals. Simulation Using SAINT: A User-Oriented Instruction Manual (1) provides an analyst with the modeling concepts necessary to build SAINT models. This manual, The SAINT User's Manual, contains all of the information necessary to use the SAINT simulation program to analyze SAINT models. It is designed to be used after the SAINT modeling concepts are understood. Documentation for the SAINT Simulation Program (2) serves as a complete reference guide to the inner workings of the SAINT simulation program.

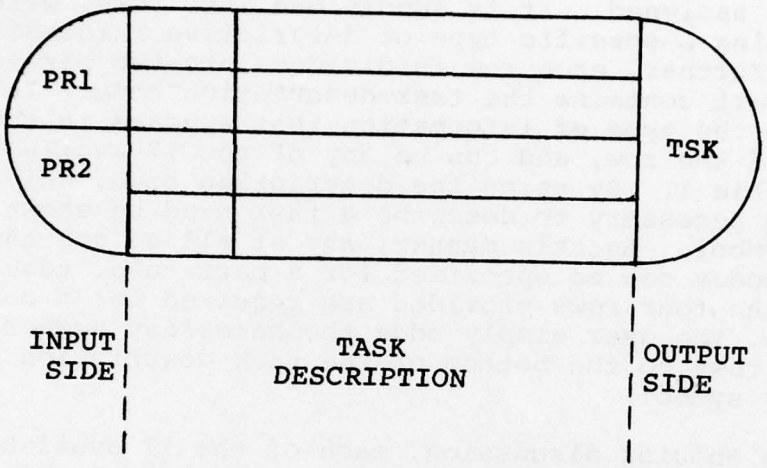
SECTION II

SAINT SYMBOLISM

The basic element of a SAINT network is a task. The symbol used to portray tasks in a SAINT model is shown in Figure 1. Three basic functions are associated with each task in a SAINT network: 1) release of the task (satisfying all predecessor requirements); 2) performance of the task and associated operations; and 3) identification of potential successor tasks. These three functions are described by an input side, a task description, and an output side. Each portion of the task symbol will now be described.

Task Input

The number and nature of predecessor requirements for the release of a task are shown on the input side of the task symbol. The number of predecessor completions required for first release of the task (PR1) is at the top, while the number of predecessor completions required for subsequent release of the task (PR2) is at the bottom.



TSK	task number
PR1	number of predecessor completions required for first release of the task
PR2	number of predecessor completions required for subsequent release of the task

Figure 1. Task Symbol.

Task Description

The center portion of the task symbol shown in Figure 1 contains all task description information; such as performance time characteristics, statistics to be collected, and attributes to be assigned. It is subdivided into rows, with each row containing a specific type of descriptive information about the task. Further, each row is divided into two parts. The left-hand part contains the task description code. It is used to identify the type of information that appears in the right-hand part of the row, and can be any of the 17 available codes shown in Table 1. By using the description code, only the information necessary to describe a task need be shown on the task symbol. In this manner, any or all of the task description codes can be specified for a particular task. If more than the four rows provided are required for a complete description, the user simply adds the necessary number of additional rows to the bottom of the task description portion of the task symbol.

In the ensuing discussion, each of the 17 available task description codes will be described. The type of information that is identified by the code will be defined. In addition, examples will be shown that demonstrate the use of each code in describing a task. However, remember that the examples are only intended as guidelines. The SAINT user should employ a format that is convenient for his modeling and communication purposes. Caution should be exercised in developing new formats, as it is desirable that other SAINT modelers be able to read the network and understand the model. Many of the task description rows correspond to SAINT input cards. Thus, a trade-off exists between the ease of preparing SAINT input cards from the network model and the ease of communicating the characteristics of the SAINT model to others.

ATAS

The code ATAS identifies a row containing the attribute assignments to be made at the task. For example, if information attribute 2 is to be assigned a value drawn from distribution set 3 at the time the task is completed, then the ATAS row could appear as

ATAS	COM: IA,2=DS,3

However, if the value of attribute 4, resource 7, is to be

TABLE 1

TASK DESCRIPTION CODES

Description Code	Information Required by Description Code
ATAS	attribute assignments to be performed
DIFF	different predecessor option
DMOD	distribution set modifications to be made
INCM	information choice mechanism decision mode
LABL	label associated with task
MARK	marking information
MODF	moderator functions to be applied to task performance
PREC	completion precedence
PRTY	priority
RCLR	resource clearing to be performed
REGL	regulation to be performed
RESR	resource requirements
STAT	statistics to be collected
SWIT	switching to be performed
TCLR	task clearing to be performed
TIME	performance time characteristics
UTCH	user-defined task characteristics

assigned a value calculated in user function 11 at the start of the task, then the ATAS row could appear as

ATAS	START: RA,7-4=UF,11

If both assignments described above are to be made when the task is released, then the ATAS row could appear as

ATAS	REL: (IA,2=DS,3),(RA,7-4=UF,11)

DIFF

The code DIFF identifies a row containing the different predecessor requirements of the task. If different predecessor completions are required in order for the task to be released, then the DIFF row could appear as

DIFF	YES

If different predecessors are not required, the DIFF row could be left out of the task description or could appear as

DIFF	NO

DMOD

The code DMOD identifies a row containing the distribution set modifications to occur upon completion of the task. If distribution set 3 is to be replaced by distribution set 5, then the DMOD row could appear as

DMOD	3 → 5

or as

DMOD	DS,3 → DS,5

INCM

The code INCM identifies a row containing the information choice mechanism decision mode. If the information packet arriving with the first predecessor completion is to be saved, then the INCM row could appear as

INCM	FIRST

If the information packet with the smallest value of attribute 2 is to be saved, then the INCM row could appear as

INCM	SMALL,2

LABL

The code LABL identifies a row containing the label assigned to the task. On the network, the task label can be of any length. However, only eight alphanumeric characters are recognized by the SAINT input processor.

Assume that a task represents the action of traveling to a desired location. A task description row could then appear as

LABL	TRAVEL TO NEW LOCATION

MARK

The code MARK identifies a row containing marking information. If marking is to take place at the time of release of the task, then the MARK row could appear as

MARK	RELEASE

If the marking is to take place upon completion of the task, then the MARK row could appear as

MARK	COMPLETION

or as

MARK	COM

MODF

The code MODF identifies a row containing the moderator functions to be applied to task performance. For example, if moderator function 1 represents a speed function and moderator function 5 represents an environmental-related function, and both are to be applied to this task, then the MODF row could appear as

MODF	1,5

or as

MODF	1(SPEED), 5(ENVIRONMENT)

PREC

The code PREC identifies a row containing the task completion precedence value. If two tasks are completed at the same time, the task with the higher PREC value will be processed first. If the completion precedence is 15, then the PREC row could appear as

PREC	15.

PRTY

The code PRTY identifies a row containing the task priority used for scheduling purposes. If the priority of the task is 5, then the PRTY row could appear as

PRTY	5.

RCLR

The code RCLR identifies a row containing the resources to be cleared and the associated tasks to be signaled upon task completion. If resource 1 is to be cleared and task 53 is to be signaled, and resource 5 is to be cleared and no task signaled, then the RCLR row could appear as

RCLR	(CLR 1,SIG 53),(CLR 5)

or as

RCLR	(C1,S53),(C5,NO SIGNAL)

REGL

The code REGL identifies a row containing the state variable regulation to be performed upon task completion. If the value of SS(16) is to be set to 5.*SS(4)+3., then the REGL row could appear as

REGL	SS(16) → 5.*SS(4)+3.

If the value of SS(16) is to be decreased by the same regulating

function, then the REGL row could appear as

REGL	SS(16) ↓ 5.*SS(4)+3.

RESR

The code RESR identifies a row containing the resource requirements of the task. If the task requires resources 1, 5, and 7 for performance, then the RESR row could appear as

RESR	AND: 1,5,7

or as

RESR	AND: 1(PLUMBER),5(HAMMER),7(WRENCH)

If the task requires only one resource, and it must be either resource 2 or resource 3, then the RESR row could appear as

RESR	OR: 2,3

STAT

The code STAT identifies a row containing the statistics to be collected at the task. For example, assume that statistics are to be collected on the time between starts of the task. Further assume that the histogram to be generated contains 25 cells, with the upper limit of the first cell being 20 and the cell width being 15. Then the STAT row could appear as

STAT	BETWEEN STARTS

or as

STAT	BET STA

or as

STAT	BET STA (25.,20.,15.)

SWIT

The code SWIT identifies a row containing the switching to be performed upon task completion. If the value of switch 3 is to be set to 152, then the SWIT row could appear as

SWIT	IS,3=152

or as

SWIT	SET SWITCH 3 TO 152

TCLR

The code TCLR identifies a row containing the tasks to be cleared and the associated tasks to be signaled upon task completion. For example, if task 27 is to be cleared and task 39 is to be signaled upon completion of this task, then the TCLR row could appear as

TCLR	27,39

or as

TCLR	CLEAR 27,SIGNAL 39

TIME

The code TIME identifies a row containing the performance time characteristics. If the task time is located in distribution set 3, then the TIME row could appear as

TIME	DS,3

or as

TIME	DS,3 (NORMAL,MEAN=5.75)

If the task performance time is located in attribute 5 of resource 1, then the TIME row could appear as

TIME	RA,1-5

UTCH

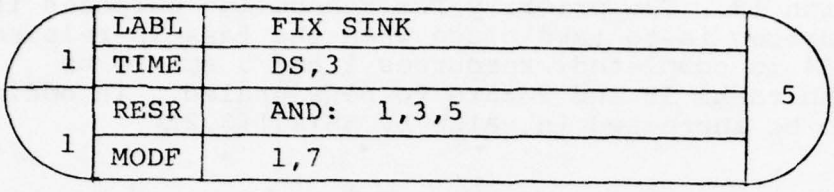
The UTCH code identifies a row containing the user-defined task characteristics of the task. If the task requires user-defined task characteristics 1, 5, and 6; and they have values of 2.75, 1.9, and 2, respectively; then the UTCH row could appear as

UTCH	(1,2.75),(5,1.9),(6,2.)

Examples of Task Descriptions

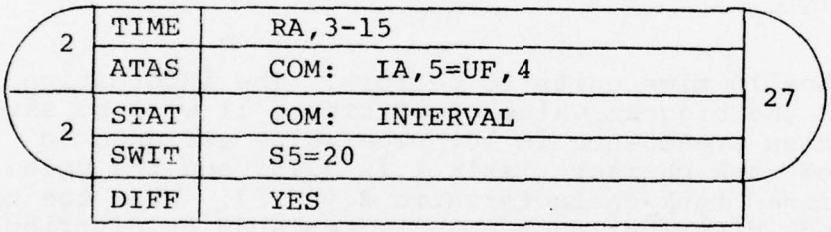
In the previous discussion, definitions and examples of the 17 available description codes were presented. The following examples show how the individual description rows can be integrated to form a complete task description.

Example 1



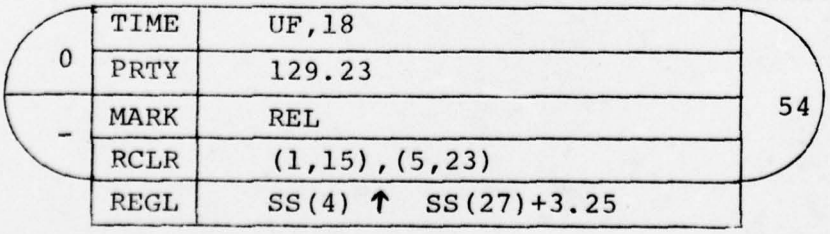
Task 5 represents the operation of fixing a sink. The baseline time to perform the task is a deviate drawn from distribution set 3. It is to be performed with resources 1, 3, and 5. Moderator functions 1 and 2 are to be applied to task performance.

Example 2



The time to perform task 27 is located in attribute 15 of resource 3. Upon completion of the task, information attribute 5 is to be assigned a value calculated in user function 4. Interval statistics (time since the last marking) are to be collected upon task completion. Finally, the value of switch 5 is to be set to 20. In addition, different predecessor completions are required in order for the task to be released.

Example 3



The time to perform task 54 (a source task, since it requires no predecessor completions for release) is calculated in user function 18. The priority for scheduling purposes is 129.23. Marking is to take place when the task is released. When task 54 is completed, resources 1 and 5 are to be cleared, and tasks 15 and 23 are to be signaled. In addition, SS(4) is to be increased in value by SS(27)+3.25.

Example 4

10	TIME	SC,10	17
	INCM	BIG,11	
100	PREC	10.	
	UTCH	(1,2.75),(2,157.)	
	DMOD	DS,10 → DS,21	
	TCLR	C35,S14	

Task 17 takes 10 time units to perform. The information packet with the biggest value of attribute 11 will be saved. The completion precedence is 10. The value assigned to user-defined task characteristic 1 is 2.75, and the value of user-defined task characteristic 2 is 157. When the task is completed, distribution set 10 is replaced by distribution set 21, task 35 is cleared, and task 14 is signaled.

Task Output

Each task in a SAINT network is uniquely identified by a task number shown on the output side of the task symbol. The shape of the output side indicates the type of branching operation to be performed upon task completion (deterministic, probabilistic, conditional-take first, or conditional-take all). The symbols for the four types of branching operations available in SAINT are shown in Table 2. Tables 3 and 4 define and present examples of the branch codes for probabilistic and conditional branching. Deterministic branching requires no branch codes.

TABLE 2

AVAILABLE BRANCHING TYPES

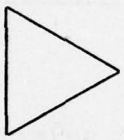
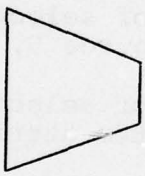
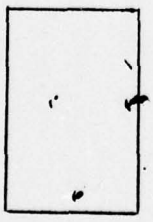
Branching Type	Symbol	Description of Branching Operation
Deterministic		All branches emanating from this task will be selected.
Probabilistic		One and only one of the branches emanating from this task will be selected. The selection is made on a random basis using the probabilities associated with the branches.
Conditional-Take First		One and only one of the branches emanating from this task will be selected. The branches are rank ordered and the associated conditions are tested. When one of the conditions is satisfied, the associated branch is selected.
Conditional-Take All		All branches whose conditions are satisfied will be selected. As with conditional-take first branching, the branches are rank ordered and the associated conditions are tested. All branches emanating from the task are tested for possible selection.

TABLE 3

PROBABILISTIC BRANCHING CODES

Branch Code	Description
(p)	Probability of selection equals p
(T,A)	Probability of selection equals the value of attribute A of type T

Examples of the branch codes are:

(0.75)	Probability of selecting branch is 0.75
(IA,7)	Probability of selecting branch is the value of information attribute 7
(RA,5-3)	Probability of selecting branch is the value of resource 5, attribute 3
(SA,25)	Probability of selecting branch is the value of system attribute 25

TABLE 4

CONDITIONAL BRANCHING CODES

Branch Code	Description of Condition for Selecting Branch
(TLV,x)	TNOW.LE.x
(TGV,x)	TNOW.GT.x
(TLA,T,A)	TNOW.LE.v(T,A)
(TGA,T,A)	TNOW.GT.v(T,A)
(TVC,N)	Task N has been completed
(TVN,N)	Task N has not been completed
(TAC,T,A)	Task identified as v(T,A) has been completed
(TAN,T,A)	Task identified as v(T,A) has not been completed
(ALV,T,A,x)	v(T,A).LE.x
(AGV,T,A,x)	v(T,A).GT.x
(ALA,T,A,B)	v(T,A).LE.v(T,B)
(AGA,T,A,B)	v(T,A).GT.v(T,B)

Definitions of the terms used in the descriptions are:

TNOW	current simulation time (time of task completion)
.LE.	"less than or equal to"
.GT.	"greater than"
x	specified real value
v(T,A)	value of attribute A of type T
v(T,B)	value of attribute B of type T

Examples of the branch codes are:

(TLV,5.75)	Select branch if time of task completion is less than or equal to 5.75
(TGA,IA,7)	Select branch if time of task completion is greater than the value of information attribute 7
(TVC,35)	Select branch if task 35 has been completed
(TAN,RA,4-1)	Select branch if the task whose number is the value of resource 4, attribute 1, has not been completed
(ALV,SA,12,2.3)	Select branch if the value of system attribute 12 is less than or equal to 2.3
(AGA,IA,3,5)	Select branch if the value of information attribute 3 is greater than the value of information attribute 5

Additional Task Modeling Concepts

Task modification involves the substitution of the task description and output side of one task for those of another task. The illustration in Figure 2 indicates that task 8 will be replaced by task 9 upon completion of task 2 (the symbol △ identifies a task modification operation).

Task and resource clearing involve the interruption (clearing) of a task in progress when another task is completed. If a clearing operation is performed, SAINT allows the number of predecessor requirements for another task to be reduced by one (signaling). As indicated previously, the clearing information is included as a part of the description of the task whose completion causes the clearing. The illustration in Figure 3 indicates that task 37 is to be signaled as a result of the clearing caused by the completion of task 5 (the symbol ○ identifies a task signaling operation caused by a task or resource clearing).

SAINT also allows the number of predecessor requirements for a task to be reduced by one as a result of a threshold crossing detected by a monitor. Figure 4 illustrates a situation in which task 20 will be signaled when monitor 10 detects a threshold crossing (the symbol □ identifies a task signaling operation caused by a threshold crossing).

State Variable Definitions

Although the SAINT user must set the initial values of state variables in subroutine INTLC and code their defining equations in subroutine STATE; symbolism is available to define state variables on the network model. For each state variable included in a SAINT model, four pieces of information are defined: state variable number, state variable label, defining equation in STATE, and initial value in INTLC. State variable information is depicted on the network model in the manner shown in Figure 5.

Monitors

The symbol which is used to represent a state variable monitor in a SAINT model is shown in Figure 6. Its design is similar to that provided for task descriptions. The right-hand side of the symbol contains the monitor description, and is divided into four rows. Each row represents a unique type of descriptive information, and is identified by one of the following description codes: LABL, MONF, MTAS, and MSWT. By using the description codes, only the information necessary to describe a monitor need be shown on the monitor symbol. Each code will be defined and examples of their use presented in the following discussion.

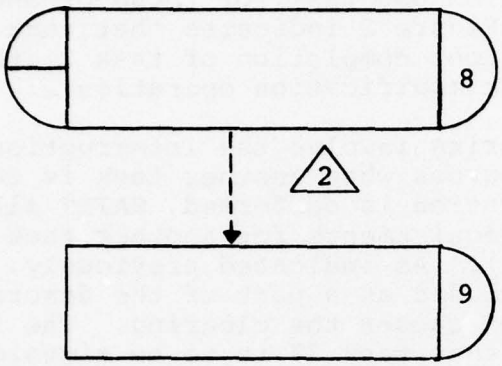


Figure 2. Task Modification.

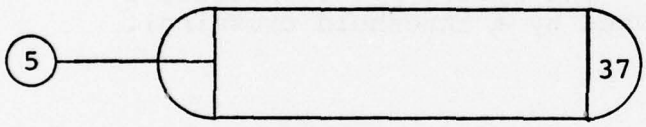


Figure 3. Task Signaling as a Result of Task or Resource Clearing.

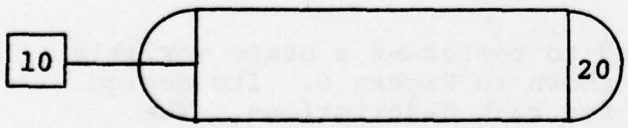


Figure 4. Task Signaling as a Result of Threshold Crossing.

State Variable Number	State Variable Label	Defining Equation in STATE	Initial Value in INTLC
1	PRESSURE	DD(1)=SS(3)	5.2
2	ACCELERATION	DD(2)=4.*SS(1)	0.
3	OIL LEVEL	-	5.
4	STRESS	SS(4)=SSL(4)+DTNOW*(1-LEVEL)	7.39
5	POSITION	SS(5)=SSL(5)+POS*DTNOW	27.5

Figure 5. State Variable Information on Network Model.

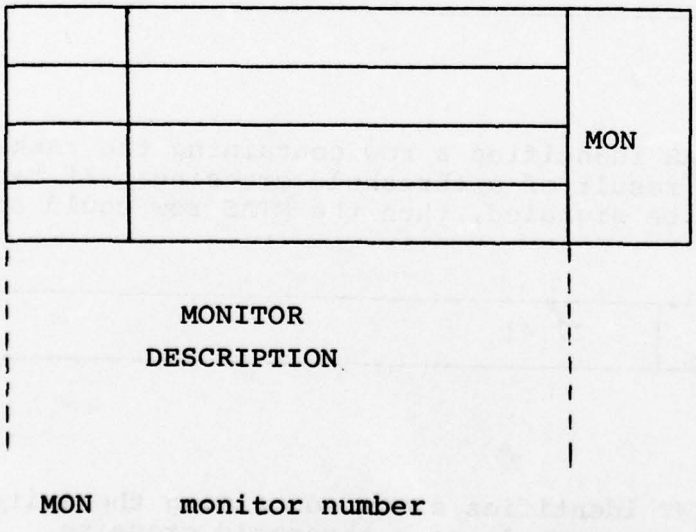


Figure 6. State Variable Monitor Symbol.

LABL

The code LABL identifies a row containing the monitor label. For example, if pressure is being monitored, then the LABL row could appear as

LABL	PRESSURE

MONF

The code MONF identifies a row containing the function to be used in locating the threshold crossing. For example, suppose that the value of SS(1) is being monitored. A threshold crossing occurs when the value of SS(1) falls below the value of 3.0*SS(7)+4.0 within a tolerance of 0.05. The MONF row could appear as

MONF	SS(1) ↓ 3.0*SS(7)+4.0(TOL=.05)

or as

MONF	SS(1) CROSSING BELOW 3.0*SS(7)+4.0 WITHIN 0.05

MTAS

The code MTAS identifies a row containing the tasks to be signaled as a result of a threshold crossing. If tasks 27 and 41 are to be signaled, then the MTAS row could appear as

MTAS	27,41

MSWT

The code MSWT identifies a row containing the switch values to be set as a result of a threshold crossing. If the value of switch 3 is to be set to 27, then the MSWT row could appear as

MSWT	IS,3=27

or as

MSWT	SET SWITCH 3 TO 27

Example of Monitor Description

In the previous discussion, definitions and examples of the four available description codes were presented. The following example shows how the individual description rows can be integrated to form a complete monitor description.

LABL	SOLUTION CONCENTRATION	15
MONF	SS(3) ↑ SS(10)+17.2,TOL-0.5	
MTAS	35	
MSWT	(IS,1-3),(IS,5=40)	

The example shown above is the symbol describing monitor 15. The state variable being monitored, SS(3), is the concentration of a solution. When the concentration increases beyond the value of SS(10)+17.2, with a tolerance of 0.5, a threshold crossing occurs. This causes task 35 to be signaled, switch 1 to be set to 3, and switch 5 to be set to 40.

This concludes the discussion of monitor symbolism. Note that only those rows necessary to completely describe the monitor need be shown on the monitor symbol.

SECTION III

SAINT DATA INPUT PROCEDURE

The SAINT simulation program is a FORTRAN program for simulating SAINT models. The user of SAINT need only master the procedures for translating a model into data input cards in order to obtain an analysis of the model. This section specifies the procedures involved in this translation. It provides the complete specifications for preparing the input to the SAINT program. A list of the SAINT data types is presented in Table 5. The procedures for data input card preparation are described next. This is followed by a detailed description of each data type and their corresponding input fields. A condensed version of the data type descriptions is included at the end of this section for the user who is familiar with the detailed description.

Data Input Card Preparation

All SAINT data input is prepared in free format. Free format input permits information to be punched without card column restrictions, where each input variable is defined as a field of the data input card. The features of SAINT free form input include:

1. Card ID - Free form data cards are uniquely identified by an ID specified in the first field of the card. Only the first three characters on this ID are significant. Valid card types are identified in Table 5.

2. Blanks - Leading blanks and blanks imbedded in numeric fields are ignored.

3. Field Termination - All fields on a card except the last are terminated by commas.

4. Continuation Cards - Continuation cards are permitted. If the last non-blank character of a card is a comma, it is assumed that additional fields of the input record are contained on the card which follows. Fields may not be split between cards. A continuation card contains no card ID and the additional fields may be punched anywhere in the card. Continuation cards may themselves be continued. However, an input record may not exceed 50 fields.

5. Record Termination - An asterisk should be punched after the last field of an input card. If no asterisk is present and the last non-blank character in the

TABLE 5

LIST OF SAINT INPUT DATA TYPES

General Information

GEN	General Information
SGE	State variable general information
POP	Program options
OUT	Output options
DIS	Distribution set definitions
SIM	Used when multiple networks are to be simulated to mark the end of input for each network except the last
FIN	Signals end (finish) of all SAINT input cards

User-Generated Statistics and Plots

UBO	User-generated statistics for variables based on observation
UTI	User-generated statistics for time-persistent variables
UHI	User-generated histograms
UPL	User-generated plot/table characteristics
UVA	User-generated plot/table variable information

Initial Status and Labels

IMO	Initial moderator function status specification
IRA	Initial resource attribute specification
ISA	Initial system attribute specification
LRE	Labels for resources
LSV	Labels for state variables

Task Description

TAS	Task characteristics
STA	Statistics task information
UTC	Specification of user-defined task characteristics
MOD	Specification of moderator function status
ATA	Specification of attribute assignment information

Task Branching

DET	Deterministic branching
PRO	Probabilistic branching
CFI	Conditional-take first branching
CAL	Conditional-take all branching

Effects of Task Completions

NMO	Network modifications
DMO	Distribution set modifications
TCL	Task clearing
RCL	Resource clearing
SWI	Switching information
REG	Regulation information

State Variable Monitors

MON	Monitor characteristics
MTA	Monitor actuated task signaling
MSW	Monitor actuated switching

State Variable Statistics and Plots

SST	State variable statistics information
PLO	State variable plot/table characteristics
VAR	State variable plot/table variable information

card is not a comma, then an end of input record is assumed. The use of an asterisk is preferred since it reduces scanning time.

6. Alphanumeric Information - Alphanumeric data may begin with a letter or a digit and may contain any combination of letters and digits. In any alphanumeric field, characters which are not significant or required for that field may be included for clarity and documentation purposes, but will be ignored in processing. Alphanumeric fields contain at most 8 significant characters.

7. Numeric Information - Any numeric information may be input as an integer or as a real number.

   a. If an integer is input for a field specified as real, the real equivalent of the integer value is used.

   b. If a real is input for a field specified as integer, the decimal portion of the real field is truncated and the integer equivalent of the truncated result is used.

   Real values may also be input as E-format ("floating-point") numbers. A valid E-format field consists of a number followed by the letter E and a one or two digit power of 10. The number may be signed or unsigned. The exponent must not contain a decimal point and may be signed or unsigned. (Largest valid exponent is machine dependent.) Integers and E-format reals are not interchangeable. A warning message will be generated if an E-format real is submitted for a field which should contain an integer.

   Examples:

   ```
   +53.9E+10
   53.9E10
   53.9 E-10
   .539E -8
   1.E20
   -1.0E+20
   ```

8. Default Values - Default values are defined for all nonessential input fields. To indicate that the default value is to be used for a certain field (or that a field is not applicable in a given context), the user should do one of the following:

   a. Omit the field - Omission of a field is

indicated by a comma or by blanks followed by a comma.

b. Skip to the next user-specified input field - If the user lists the number (enclosed in parentheses) of the next field for which he wishes to specify information, all intermediate fields will be bypassed and will assume default values. For example, if the following card is input

```
POP,4,,5,(8)MULTIPLE*
```

these assumptions will be made:

```
field 1 = POP
field 2 = 4
field 3 will assume a default value
field 4 = 5
fields 5 through 7 will assume default
    values
field 8 = M
field 9 will assume a default value
```

c. Terminate the card before giving a value for the field. For instance, in the preceding example field 9 assumes a default value since nine fields are associated with card type POP and only eight values were specified on the input card.

9. Deck Setup - Cards for a single network may be input in any order with the following exceptions:

a. GEN, SGE (if any), POP, OUT, and DIS cards must be input before any other card types.

b. VAR cards must follow their associated PLO card. Similarly, UVA cards must follow their associated UPL card.

c. A SIM or a FIN card must be the last input card for a network. (A SIM card is used if input for an additional network follows.)

Only those cards which will be used for a particular network need to be input. For example, if no network modifications or task clearings are included in the model, no NMO or TCL cards should be submitted.

SAINT networks can be sequentially analyzed by stacking the data cards for each network one after another. A SIM card should separate the data cards for each network.

Detailed Description of Data Types

Each data type will not be presented. For each field in a data card, the field number, the nature of the input (A=alphanumeric, I=integer, R=real), the associated default values (if any), and a description of field contents will be listed. In the specification of alphanumberic inputs, the required characters for each code are underlined in the description of the code if not all characters shown are required.

GEN - General Information

Field	Value	Default	Description
1	A	-----	Card identification (GEN).
2	A	SAINT	User name (maximum of 8 characters).
3	I	1	Month.
4	I	1	Day.
5	I	2001	Year.
6	I	1	Project number.
7	I	1	Number of iterations.
8	I	1	Number of sink tasks that must be completed for network to be completed.
9	I	71268659	Integer random number seed.
10	R	1.	Scale factor to be used in conjunction with function type SC.
11	A	Y	Print suppression key: Yes = Print echo check. No = Skip printing of echo check.

SGE - State Variable General Information

This data type is input only if state variables are defined for the simulation.

Field	Value	Default	Description
1	A	-----	Card identification (SGE).
2	I	0	Number of equations (largest index) defining state variables written in terms of DD(·). (NNEQD)
3	I	0	Number of equations defining state variables written in terms of SS(·). (NNEQS)
4	R	(.01*DTMAX)	Minimum step size permitted (DTMIN).
5	R	(Min(DTSAV,1.E20))	Maximum step size permitted (DTMAX).
6	R	1.E20	Communication interval for recording values on plots/tables (DTSAV) >0 Time between communication points =0 Communication to occur at each accepted update point <0 Communication only at event times
7	R	.00001	Absolute local truncation error allowed in Runge-Kutta integration. (AAERR)
8	R	.00001	Relative error allowed in Runge-Kutta integration. (RRERR)
9	A	W	Key to indicate severity of error when accuracy cannot be maintained by the Runge-Kutta integration package, either in state variable updates or searching for threshold crossings. None - Proceed without warning message Warning - Proceed with warning message Fatal - Fatal error Accuracy is required to be less than or equal to absolute error + (relative error *SS(I)) for all I.

POP - Program Options

Field	Value	Default	Description
1	A	-----	Card identification (POP).
2	I	0	Number of resources involved in the simulation (largest resource number).
3	I	0	Number of resource attributes desired per resource packet.
4	I	0	Number of information attributes desired per information packet.
5	I	0	Number of system attributes desired.
6	I	0	Number of moderator functions desired.
7	A	I	Network Modifications (if any) Interchange - Interchange modifications desired Multiple - Multiple replacement modifications desired
8	A	I	Distribution Set Modifications (if any) Interchange - Interchange modifications desired Multiple - Multiple replacement modifications desired.
9	I	3	Ranking procedure for the file of tasks awaiting scheduling 1 Low value first on priority 2 High value first on priority 3 First-in, First out (FIFO) 4 Last-in, Last-out (LILO)

OUT - Output Options

Field	Value	Default	Description
1	A	-----	Card identification (OUT).
2	I	0	Iteration number for which the detailed output should begin.
3	I	value in field 2	Iteration number for which the detailed iteration output should terminate.
4	I	0	Iteration number for which the resource utilization summary output should begin.
5	I	value in field 4	Iteration number for which the resource utilization summary output should terminate.
6	I	0	Iteration number for which the statistics task summary output should begin.
7	I	value in field 6	Iteration number for which the statistics task summary output should terminate.
8	I	0	Iteration number for which initial and final state variable value outputs should begin.
9	I	value in field 8	Iteration number for which initial and final state variable value outputs should terminate.
10	I	0	Iteration number for which state variable statistical output should begin.
11	I	value in field 10	Iteration number for which state variable statistical output should terminate.
12	I	0	Iteration number for which the preparation of state variable plots/tables should begin.

Field	Value	Default	Description
13	I	value in field 12	Iteration number for which the preparation of state variable plots/tables should terminate.
14	A	Y	Print suppression key: Yes = Print resource utilization summary report. No = Do not print report.
15	A	Y	Print suppression key: Yes = Print statistics task summary report. No = Do not print report.
16	A	N	Print suppression key: Yes = Print histograms of statistics tasks values from iteration 1. No = Do not print histograms.
17	A	N	Print suppression key: Yes = Print histograms of statistics tasks values from summary report. No = Do not print histograms.

DIS - Distribution Set Definitions

All distribution sets required for task performance times and attribute assignments are defined by these input cards. One card is required for each different distribution set number.

Field	Value	Default	Description
1	A	-----	Card identification (DIS).
2	I	-----	Distribution set number.
3	A	CO	Distribution type (maximum of 2 characters).
4-7	R	0.	Parameter values for the distrition specified in Field 3 as defined by the following input description.

Available SAINT Distributions

In SAINT, sample values can be obtained from any of the following 11 distributions:

BE Beta

BP Beta fitted to 3 values as in PERT

CO Constant

ER Erlang (includes Exponential)

GA Gamma

LO Lognormal

NO Normal

PO Poisson

TR Triangular

UN Uniform

WE Weibull

Parameters Required for SAINT Distributions

In SAINT, samples are obtained from the distributions such that if a sample is less than the minimum value, the sample value is assigned the minimum value. Similarly, if the sample is greater than the maximum value, the sample value is assigned the maximum value. This is not sampling from a truncated distribution, but sampling from a distribution with a given probability of obtaining the minimum and maximum values.

The parameters required to sample from the distributions available in SAINT are described below. Fields 1, 2, and 3 are the same for all distributions.

For distribution type CO (Constant):

Field 4 The constant time

For distribution type NO (Normal); LO (Lognormal); BE (Beta); and GA (Gamma):

Field 4 The mean value
Field 5 The minimum value
Field 6 The maximum value
Field 7 The standard deviation value

For distribution type UN (Uniform):

Field 4 Not used
Field 5 The minimum value
Field 6 The maximum value
Field 7 Not used

For distribution type ER (Erlang):

Field 4 The mean time for the Erlang variable divided by the value given to Field 7
Field 5 The minimum value
Field 6 The maximum value
Field 7 The number of exponential deviates to be included in the sample obtained from the Erlang distribution

If Field 7 is set equal to 1, an exponential deviate will be obtained from distribution type ER.

For distribution type PO (Poisson):

Field 4 The mean minus the minimum value
Field 5 The minimum value
Field 6 The maximum value
Field 7 Not used

Care is required when using the Poisson since it is not usually used to represent an interval of time.

For distribution type BP (Beta fitted to 3 values as in PERT); TR (Triangular):

Field 4 The most likely value, m
Field 5 The optimistic (smallest) value, a
Field 6 The pessimistic (largest) value, b
Field 7 Not used

For distribution type WE (Weibull):

Field 4 α ($\alpha > 0$)
Field 5 The minimum value
Field 6 The maximum value
Field 7 β ($\beta > 0$)

where:

$$f_x(x) = \alpha\beta x^{\beta-1}\exp[-\alpha x^{\beta}], \text{ and}$$

$$F_x(x) = 1 - \exp[-\alpha x^{\beta}]$$

UBO - User-Generated Statistics for Variables Based on Observation.

This card type is required for user-generated statistics for variables based on observation.

Field	Value	Default	Description
1	A	-----	Card identification (UBO).
2	I	-----	Statistic number (used as an argument to subroutine UCLCT).
3	A	"-----"	Statistic label (maximum of 8 characters).

Fields 2 and 3 may be repeated so that up to 24 statistics may be specified in a single input record. Additional UBO cards may also be used.

UTI - User-Generated Statistics for Time Persistent Variables

This card type is required for user-generated statistics for time-persistent variables.

Field	Value	Default	Description
1	A	-----	Card identification (UTI).
2	I	-----	Statistic number (used as an argument to subroutine UTMST OR UTMSA).
3	A	"-----"	Statistic label (maximum of 8 characters).
4	R	0.	Initial value of the variable to be observed.

Fields 2 - 4 may be repeated so that up to 16 statistics may be specified in a single input record. Additional UTI cards may also be used.

UHI - User-Generated Histograms

This card type is required for user-generated histograms.

Field	Value	Default	Description
1	A	-----	Card identification (UHI).
2	I	-----	Histogram number (used as an argument to subroutine UHIST).
3	A	"-----"	Histogram label (maximum of 8 characters).
4	I	25	Number of cells in histogram.
5	R	0.	Upper limit of first cell of histogram.
6	R	1.	Width of each cell of histogram.

Fields 2 - 6 may be repeated so that up to 9 histograms may be specified in a single input record. Additional UHI cards may also be used.

User-Generated Plots/Tables

User-generated plot/table information is contained on two types of data cards. The first card (UPL card) contains information on the plot/table. The second card (UVA card) contains information on the variables to be plotted/tabled. One or more UVA cards may be input to describe the variables to be included in the plot/table described by UPL card. One UPL card and its associated UVA cards are required for each user-generated plot/table.

UPL - User-Generated Plot/Table Characteristics

Field	Value	Default	Description
1	A	-----	Card identification (UPL).
2	I	-----	Plot/table number (used as an argument to subroutine UPLOT).
3	A	"TIME"	Independent axis label (maximum of 8 characters).
4	A	0	Storage medium to be used for plot/table values: > 0 Peripheral storage unit number on which plot/table information is to be written (required if multiple plots/tables are desired). = 0 Plot/table information to be stored in core (UPSET(·)). Only 1 plot/table can be requested and Field 2 must be set to 1 if core is used.
5	A	P	Plot/table option Quick - Quick Plot Plot - Plot only Table - Table only Both - Plot and table.
6	R	5.	Increment of the independent variable between successive plot points.

UVA - User-Generated Plot/Table Variable Information

Field	Value	Default	Description
1	A	-----	Card identification (UVA).
2	I	-----	Plot/Table Number (same as Field 2 of associated UPL card).
3	I	-----	Variable number.
4	I	value in field 3	Plot symbol (maximum of 1 character).
5	A	"-----"	Label associated with the variable being plotted/tabled (maximum of 8 characters).
6	A	M	Key for specifying the lower limit of the scale to be used for the variable Minimum - Use minimum value observed during the simulation. Value - Use value in Field 8. Rounded - Use the largest multiple of the value specified in Field 8 that is less than or equal to the minimum value observed.
7	A	M	Key for specifying the upper limit of the scales to be used for this variable Maximum - Use maximum value observed during the simulation. Value - Use value in Field 9. Rounded - Use the smallest multiple of the value specified in Field 9 that is greater than or equal to the maximum value observed.
8	R	0.	Value associated with the lower limit of the plot ordinate (see Field 6).

Field	Value	Default	Description
9	R	0.	Value associated with the upper limit of the plot ordinate (see Field 7).

Fields 3 - 9 may be repeated so that up to 6 variables may be described on a single data card. Additional UVA cards may also be used to describe additional variables.

IMO - Initial Moderator Function Status Specification

Field	Value	Default	Description
1	A	-----	Card identification (IMO).
2	I	-----	Moderator function number.
3	A	I	Initial moderator function status Inactive - Moderator function inactive Active - Moderator function active.

Fields 2 and 3 may be repeated so that the initial status of up to 24 moderator functions may be specified on a single data card. Additional IMO cards may also be used to initialize additional moderator functions.

If the initial status of a moderator function is not specified by this data type, it is assumed to be inactive.

IRA - Initial Resource Attribute Specification

This card is used to specify the initial resource attribute assignments. All assignments specified on a single data card are assigned to the resource whose number is specified in Field 2 of that card.

Field	Value	Default	Description
1	A	-----	Card identification (IRA).
2	I	-----	Resource number associated with all assignments specified on this card.
3	I	-----	Resource attribute number to which the assignment specified in Fields 4 and 5 is to be made.
4	A	SC	The function to be used in conjunction with the parameter specification (Field 5) in making the assignment. The allowable functions are: DS Value sampled from the distribution set specified as the parameter specification. SC Constant equal to the parameter specification divided by the scale factor (Field 21 of GEN card). UF User-written function (used in conjunction with function USERF (I) where I is the parameter specification).
5	I	0	Parameter specification to be used in making the assignment.

Fields 3 - 5 may be repeated so that up to 10 initial resource attribute assignments can be made on a single data card. Additional IRA cards may also be used to make additional initial resource attribute specifications.

ISA - Initial System Attribute Specification

Field	Value	Default	Description
1	A	-----	Card identification (ISA).
2	I	-----	System attribute number to which the assignment specified in Fields 3 and 4 is to be made.
3	A	SC	The function to be used in conjunction with the parameter specification (Field 3) in making the assignment. The allowable functions are: DS Value sampled from the distribution set specified as the parameter specification SC Constant equal to the parameter specification divided by the scale factor (Field 12 of GEN card) UF User-written function (used in conjunction with function USERF(I), where I is the parameter specification).
4	I	0	Parameter specification to be used in making the assignment.

Fields 2 - 4 may be repeated so that up to 10 initial system attribute assignments can be made on a single data card. Additional ISA cards may also be used.

LRE - Labels for Resources

This card type is required to assign labels to resources.

Field	Value	Default	Description
1	A	-----	Card identification (LRE).
2	I	-----	Number of the resource to be assigned a label.
3	A	"-----"	Alphanumeric label for the resource specified in Field 2 (maximum of 8 characters).

Fields 2 and 3 may be repeated so that a total of 10 resource labels can be specified in a single input record. Additional LRE cards may also be used.

LSV - Labels for State Variables

This card type is required to assign labels to state variables.

Field	Value	Default	Description
1	A	-----	Card identification (LSV).
2	I	-----	Number of the state variable to be assigned a label.
3	A	"-----"	Alphanumeric label for the state variable specified in Field 2 (maximum of 8 characters).

Fields 2 and 3 may be repeated so that a total of 10 state variable labels can be specified in a single input record. Additional LSV cards may also be used.

Task Definitions

Information about a task is contained on up to six data cards. The first card (TAS card) contains information about the characteristics of the task. The second card (STA card) contains information concerning marking and the statistics to be collected at the task. This card is only used if marking is to be performed or statistics are to be collected at the task. The third card (UTC card) contains the values of user-defined task characteristics. This card is only used if the number of user-defined task characteristics is greater than zero. The fourth card (MOD card) contains moderator function status information. This card is only used if any status changes are to be made. The fifth card (ATA card) contains attribute assignment information. This card is only used if attribute assignments are to be made. The sixth card contains information about the task's branching and successor tasks. A branching card is only used if branching is performed from the task. The following branching types are allowed:

deterministic (DET card)
probabilistic (PRO card)
conditional-take first (CFI card)
conditional-take all (CAL card)

TAS - Task Characteristics

Field	Value	Default	Description
1	A	-----	Card identification (TAS).
2	I	-----	The number of the task whose characteristics are given on this card.
3	A	"-----"	Task Label (maximum of 8 characters).
4	I	0 if source task, 1 otherwise	The number of predecessor completions required to release the task for the first time.
5	I	9999	The number of predecessor completions required for all subsequent releases of the task.
6	A	SC	The function to be used in conjunction with the parameter specification in Field 7 in obtaining the task performance time. The allowable functions are: DS Value sampled from the distribution set specified as the parameter specification SC Constant equal to the parameter specification divided by the scale factor (Field 12 of GEN Card) UF User-written function (used in conjunction with function USERF (I), where I is the parameter specification) IA Value taken from the information attribute specified as the parameter specification RA Value taken from the resource attribute specified as the parameter specification SA Value taken from the system attribute specified as the parameter specification.
7	I	0	Parameter specification to be used in conjunction with Field 6 in obtaining a value for the task performance time.

Field	Value	Default	Description
8	I	0	Resource number to be used if function RA was selected in Field 6.
9	R	0.	Task priority.
10	A	NS	Special task characteristics SOurce - Source task SInk - Sink task NS - No special characteristic is associated with the task.
11	A	N	Indicator that different predecessor completions are required to release the task. Yes - Different predecessor completions required No - No requirement for different predecessor completions.
12	A	L	Information choice mechanism decision mode First - Hold packet arriving with first requirement Last - Hold packet arriving with last requirement Big - Hold packet with the largest value in attribute specified in Field 13 Small - Hold packet with the smallest value in attribute specified in Field 13.
13	I	0	Information attribute to be used for comparison if mode Big or Small is selected in Field 12.
14	R	0.	Task completion precedence.
15	A	A	Resource Requirement Code And - All resources specified (if any) required for performance Or - One of the resources specified required for performance.
16-20	I	-----	Resources associated with this task (must appear in ascending numerical order; maximum of five resources permitted).

STA - Statistics Task Information

This card is required only if marking is to be performed or statistics are to be collected at this task. Only one STA card per task is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (STA).
2	I	-----	Task number.
3	A	N	Indicator that marking is to be performed at this task: M - marking N - no marking.
4	A	COM	Time of marking: FPC - first predecessor completion REL - release of task STA - start of task COM - completion of task.
5	A	NO	Statistics type: FIR - first ALL - all BET - between INT - interval NUM - number NO - no statistics.
6	A	COM	Time of statistics collection REL - release of task STA - start of task COM - completion of task CLR - clearing of task.
7	I	25	Number of cells for the histogram.
8	R	0.	Upper limit of the 1st cell of the histogram.
9	R	1.	Cell width.

UTC - Specification of User-Defined Task Characteristics

This card is required only if there are user-defined task characteristics for this task. Only one UTC card per task is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (UTC)
2	I	-----	Task number.
3-27	R	0.	The values of the user-defined task characteristics associated with this task.

The value of each user-defined task characteristic is labeled according to the field in which is appears. In other words, the value of user-defined task characteristic I is specified in Field 3, while the value of user-defined task characteristic 25 is specified in Field 27. Up to 25 user-defined task characteristics are allowed for each task.

MOD - Specification of Moderator Function Status

This card is required only if an update in the status of any moderator function is to be made at this task. Only one MOD card per task is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (MOD).
2	I	-----	Task number.
3	I	-----	Moderator function number for which the status will be updated.
4	A	D	Status of the moderator function specified in Field 3 Deactivate - Deactivate moderator function Activate - Activate moderator function.
5	A	T	Duration of the status update specified in Field 4 Task - For the task specified in Field 2 only. Perm - Permanently: until end of simulation or next status update.

Fields 3 - 5 may be repeated so that the status of up to 10 moderator functions may be updated at a single task.

ATA - Specification of Attribute Assignment Information

This card is required only if attribute assignments are to be made at this task. Only one ATA card per task is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (ATA).
2	I	-----	Task number.
3	A	COM	Attribute assignment point REL - at task release STA - at task start COM - at task completion.
4	A	IA	Type of attribute assignment SA - System attribute assignment IA - Information attribute assignment RA - Resource attribute assignment.
5	I	0	Resource number if RA specified in Field 4.
6	I	0	The attribute number to which the assignment is to be made.
7	A	SC	The function to be used in conjunction with the parameter specification (Field 8) in making the assignment. The allowable functions are: DS Value samples from the distribution set specified as the parameter specification SC Constant equal to the parameter specification divided by the scale factor (Field 12 of GEN card) UF User written function (used in conjunction with function USERF(I), where I is the parameter specification).
8	I	0	The parameter specification to be used in making the assignment.

Fields 4 - 8 may be repeated so that up to 6 attribute assignments can be made at a single task.

Successor Tasks

A successor task card is required only if branching is to be performed following the completion of this task. Only one successor task card per task is allowed.

DET - DETERMINISTIC Branching

Field	Value	Default	Description
1	A	-----	Card identification (DET).
2	I	-----	Task number.
3	I	-----	Successor task to task specified in Field 2.

Field 3 may be repeated so that up to 20 successor tasks may be specified.

PRO - PROBABILISTIC Branching

Field	Value	Default	Description
1	A	-----	Card identification (PRO).
2	I	-----	Task number.
3	A	NO	For attribute-based probabilistic branching, the attribute type to be used SA - System attribute. IA - Information attribute. RA - Resource attribute. NO - Not attribute-based probabilistic branching.
4	I	0	Resource number if RA is specified in Field 3.
5	I	-----	Successor task number.
6	R	-----	Probability of selecting the task in Field 5 as the successor to the task in Field 2 ($0 \leq$ probability < 1). For attribute-based probabilistic branching, the attribute number from which the probability is to be taken.

Fields 5 and 6 may be repeated so that up to 10 successor tasks may be specified. The first branch specified (Fields 5 and 6) is identified as the first branch from the task specified in Field 2.

CFI - CONDITIONAL Branching - Take FIRST branch for which the specified condition is satisfied.

Field	Value	Default	Description
1	A	-----	Card identification (CFI).
2	I	-----	Task number.
3	I	-----	Successor task number.
4	A	NO	Code signifying the type of condition to be met for selecting task in Field 3 as the successor to the task given in Field 2 according to the value specified in Field 5. TVC-The task specified as value of Field 5 has been completed. TVN-The task specified as value of Field 5 has not been completed. TLV-Time of completion of the task in Field 2 is less than or equal to the time specified in Field 5. TGV-Time of completion of the task in Field 2 is greater than the time specified as value of Field 5. ALV-The value of the attribute specified in Field 6 is less than or equal to the value specified in Field 5. AGV-The value of the attribute specified in Field 6 is greater than the value specified in Field 5. TAC-The task whose number is found in the attribute specified in Field 5 has been completed. TAN-The task whose number is found in the attribute specified in Field 5 has not been completed. TLA-Time of completion of the task in Field 2 is less than or equal to the time found in the attribute specified in Field 5. TGA-Time of completion of the task in Field 2 is greater than the time found in the attribute specified in Field 5. ALA-The value found in the attribute specified in Field 6 is less than or equal to the value found in the attribute specified in Field 5.

Field	Value	Default	Description
			AGA-The value found in the attribute specified in Field 6 is greater than the value found in the attribute specified in Field 5. No -No condition required; task specified in Field 3 is selected as successor (only for last condition specified).
5	R	0.	Value associated with the condition or the attribute which contains the value.
6	I	0	Attribute to be used in a comparison of two attribute values. Otherwise, this field should be left blank.
7	A	IA, if attribute required NO, otherwise	If attribute values are required for the condition specified in Field 4, the type of attribute to be used SA-System attribute. IA-Information attribute. RA-Resource attribute. NO-No attribute values required.
8	I	0	Resource number if RA specified in Field 7.

Fields 3 - 8 may be repeated so that up to 5 successor tasks may be specified. If a sixth successor is specified, the branch is taken unconditionally.

CAL - CONDITIONAL Branching - Take ALL branches for which the specified condition is satisfied. Input specification is the same as that specified directly above for Conditional Branching - Take First.

NMO - Network Modifications

Only required if network modifications exist. Only one NMO card per task is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (NMO).
2	I	-----	Task number.
3	I	-----	The number of the task to be replaced when the task given in Field 2 is completed.
4	I	-----	The number of the task to be inserted into the network in place of the task specified in preceding field when the task in Field 2 is completed.

Fields 3 and 4 may be repeated so that a total of 10 replacements may be made based on one task completion.

DMO - Distribution Set Modification

Only required if distribution set modifications exist. Only one DMO card per task is allowed.

Field	Value	Default	Description
1	A	-----	Card Identification (DMO).
2	I	-----	Task number.
3	I	-----	The distribution set number to be replaced when the task given in Field 2 is completed.
4	I	-----	The distribution set number to be inserted in place of the distribution set specified in preceding field when the task in Field 2 is completed.

Fields 3 and 4 may be repeated so that a total of 10 replacements may be made based on one task completion.

TCL - Task Clearing

Only required if clearing of tasks is desired. Only one TCL card per task is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (TCL).
2	I	-----	Task number.
3	I	-----	Task in progress to be halted (cleared) when the task specified in Field 2 is completed.
4	I	0	Task to be signaled when task in preceding field is cleared. A zero in this field indicates no task to be signaled.

Fields 3 and 4 may be repeated so that a total of 10 tasks may be cleared based on one task completion.

RCL - Resource Clearing

Only required if clearing at resources is desired. Only one RCL card per task is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (RCL).
2	I	-----	Task number.
3	I	-----	Resource to be cleared. If resource is not busy, no clearing is performed.
4	I	0	Task to be signaled when resource in preceding field is cleared. A zero in this field indicates no task to be signaled.

Fields 3 and 4 may be repeated so that a total of 10 resources may be cleared based on one task completion.

SWI - Switching Information

This data type is necessary only if switches are to be set as a result of task completion. One card of this type is required for each task causing switching. Only one SWI card per task is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (SWI).
2	I	-----	Task whose completion will cause the switching action detailed on this card.
3	I	-----	Switch number affected.
4	I	0	Switch value given to switch number in preceding field when task in Field 2 is completed.

Fields 3 and 4 may be repeated so that up to 10 switches may be affected based on one task completion.

REG - Regulation Information

One card of this type is required for each task causing regulation of state variables. Only one REG card per task is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (REG).
2	I	-----	Task number that regulates, in the prescribed manner, the state variable specified in Field 3.
3	I	-----	Index of SS(•) variable to be regulated upon the completion of the task specified in Field 2.
4	I	0	The index of the SS(•) variable involved in determining the regulation value (if this input is positive). The index of the DD(•) variable involved (if this input is negative). Zero if no SS or DD variable involved.
5	R	1.	Multiplicative constant (CMULT) applied to the variable designated in Field 4 in determining the regulation value.
6	R	0.	A constant value added to the product of the values specified in Fields 4 and 5 in determining the regulation value.
7	A	E.	Indicator specifying how the above determined regulation value will affect the variable designated in Field 2 Increase-Increase by function value. Equal-Regulate to function value. Decrease-Decrease by function value.

Fields 3 - 7 may be repeated so that up to 2 regulations may be made based on one task completion.

State Variable Monitor Information

Information about a monitor is contained in 1, 2, or 3 data cards. The MON card contains the characteristics of the monitor. One card of this type is required for each monitor. The MTA card is used only if tasks are to be signaled when the specified threshold is reached. The MSW card is used only if switches are to be set when the specified threshold is reached.

MON - Monitor Characteristics

Field	Value	Default	Description
1	A	-----	Card identification (MON).
2	I	-----	Monitor number.
3	A	"-----"	Monitor label (maximum of 8 characters).
4	I	-----	Index of the variable to be monitored > 0 SS(•) variable. = 0 Error. < 0 DD (•) variable.
5	I	0	Index of the variable used in computing the threshold > 0 SS(•) variable. = 0 None. < 0 DD(•) variable.
6	R	1.	Multiplicative constant (CMULT) to be used in the threshold function: CADD + CMULT * value of variable in Field 5.
7	R	0.	Additive constant (CADD) to be used in the threshold function: CADD + CMULT * value of variable in Field 5.
8	A	B	Direction of crossing to be observed Positive-Positive direction. Both-Both positive and negative directions. Negative-Negative direction.
9	R	0.	The tolerance within which the crossing is to be detected.

MTA - Monitor Actuated Task Signaling

This data card is used only if tasks are to be signaled as a result of a threshold crossing. Only one MTA card per monitor is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (MTA).
2	I	-----	Monitor number.
3	I	-----	Task to be signaled when a threshold crossing is detected.

Field 3 may be repeated so that up to 10 tasks may be signaled as a result of one threshold crossing.

MSW - Monitor Actuated Switching

This data card is used only if switches are to be set as a result of a threshold crossing. Only one MSW card per monitor is allowed.

Field	Value	Default	Description
1	A	-----	Card identification (MSW).
2	I	-----	Monitor number.
3	I	-----	Index of the switch to be affected.
4	I	0	Value to be assumed by the switch specified in Field 3.

Fields 3 and 4 may be repeated so that up to 10 switches may be affected as a result of one threshold crossing.

SST - State Variable Statistics Information

This card type is required for state variables to be observed for statistical purposes.

Field	Value	Default	Description
1	A	-----	Card identification (SST).
2	I	-----	Statistic number.
3	I	-----	The index of the SS(•) variable to be observed (if positive); the index of the DD(•) variable to be observed (if negative).
4	A	"-----"	Alphanumeric label to be associated with variable identified in Field 2 (maximum of 8 characters).
5	R	0.	The initial value of the variable specified in Field 2.

Fields 2 - 5 may be repeated so that a total of 10 statistics can be specified in a single input record. Additional SST cards may also be used.

State Variable Plots/Tables

Plot/table information is contained on two types of data cards. The first card (PLO card) contains information on the plot/table. The second card (VAR card) contains information on the variables to be plotted/tabled. One or more VAR cards may be input to describe the variables to be included in the plot/table described by PLO. One PLO card and its associated VAR cards are required for each plot/table.

PLO - State Variable Plot/Table Characteristics

Field	Value	Default	Description
1	A	-----	Card identification (PLO).
2	I	-----	Plot/table number.
3	A	TIME	Independent axis label (maximum of 8 characters).
4	I	0	Storage medium to be used for plot/table values > 0 Peripheral storage unit number on which plot/table information is to be written (required if multiple plots/tables are desired). = 0 Plot/table information to be stored in core (QPSET(·)). Only 1 plot/table can be requested and Field 2 must be set to 1 if core is used.
5	A	P	Plot/table option Quick-Quick plot Plot-Plot only Table-Table only Both-Plot and Table.
6	R	5.	Increment of the independent variable between successive plot points.

VAR - State Variable Plot/Table Variable Information

Field	Value	Default	Description
1	A	-----	Card identification (VAR).
2	I	-----	Plot number (same as Field 2 of associated PLO card).
3	I	-----	Variable number.
4	I	-----	Index of the variable to be plotted/tabled > 0 SS(·) variable = 0 Error < 0 DD(·) variable
5	A	value in field 3	Plot symbol (maximum of 1 character).
6	A	"-----"	Label associated with the variable being plotted/tabled (maximum of 8 characters).
7	A	M	Key for specifying the lower limit of the scale to be used for the variable specified in Field 4 Minimum-Use minimum value observed during the simulation Value-Use the value in Field 9 Rounded-Use the largest multiple of the value specified in Field 9 that is less than or equal to the minimum value observed.
8	A	M	Key for specifying the upper limit of the scale to be used for the variable specified in Field 4 Maximum-Use maximum value observed during the simulation Value-Use the value in Field 10 Rounded-Use the smallest multiple of the value specified in Field 10 that is greater than or equal to the maximum value observed.

Field	Value	Default	Description
9	R	0.	Value associated with the lower limit of the plot ordinate (see Field 7).
10	R	0.	Value associated with the upper limit of the plot ordinate (see Field 8).

Fields 3 - 10 may be repeated so that up to 6 variables may be described on a single data card. Additional VAR cards may also be used to describe additional variables.

Condensed Data Input Description

A condensed version of the SAINT data input description is presented in Table 6. This table is provided for the user who is already familiar with the detailed description presented previously and desires a brief description of each input field for his continued use.

A brief description of each data input field is provided in Table 6. Abbreviations are used frequently. If a field is required, no default values are shown. However, if an input field is not required, the default value assigned to that field by SAINT is shown in parentheses following the description of the field contents. Some default values are based on the values input in preceding or succeeding fields. For example, if the default value for field 3 is the value input for field 2, the default value for field 3 will be shown as (F2). In addition, some input fields can take one of two default values, depending on the specific situation. Default values of this type are shown as (0/1). An explanation of the conditions under which the default value is selected is provided in the detailed input description.

Many SAINT data cards allow certain fields to be repeated in order to facilitate the preparation of data input. For each data card with this characteristic, the fields that may be repeated are identified. In addition, the maximum number of times that the repeating set of fields may be included on the data card is indicated.

Once again, Table 6 is designed for the SAINT user who is familiar with and has used the detailed input description. Since many abbreviations are employed, care should be taken in using the condensed input description. When in doubt about the contents of a field, the user should refer to the detailed input description for clarification. The page number of the detailed input description for each data type is shown below the card identifier in Field 1.

TABLE 6

CONDENSED SAINT DATA INPUT DESCRIPTION

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
GEN (39)	User Name (SAINT)	Month (1)	Day (1)	Year (2001)	Project Number (1)	Number of Iters. (1)	Reqd. Sink Comps. (1)	Integer RN Seed (7126 - 8659)	SC Scale Factor (1.0)	Echo Check? (Y)							
SGE (40)	Number of DD(.) (0)	Number of SS(.) (0)	Minimum Step (.01* F5)	Maximum Step (Min(F5, 1.E20)	Commun-ication Intrvl. (1.E20)	Abs. Error (.00001)	Rel. Error (.00001)	Error Option (W)									
POP (41)	No. of Re-sources (0)	No. of Res. Atribs. (0)	No. of Info. Atribs. (0)	No. of System Atribs. (0)	No. of Moder. Funcs. (0)	Network Mods (I)	Distr. Set Mods (I)	Ranking Rule (3)									
OUT (42)	Detail Iter. Start (0)	Detail Iter. End (F2)	Resr. Util. Start (0)	Resr. Util. End (F4)	Stat. Tasks Start (0)	Stat. Tasks End (F6)	S.V. Values Start (0)	S.V. Values End (F8)	S.V. Stats. Start (0)	S.V. Stats. End (F10)	S.V. Plots Start (0)	S.V. Plots End (F12)	Resr. Util. Sumry? (Y)	Stat. Task Sumry? (Y)	First Iter. Histo? (N)	Sumry Histo? (N)	
DIS (44)	Distr. Set Number	Distr. Type (CO)	Param. 1 (0.)	Param. 2 (0.)	Param. 3 (0.)	Param. 4 (0.)											
SIM	Used only if multiple networks are to be simulated Marks end of each network to be simulated except the last.																
FIN	Marks end of last network to be simulated																
UBO (47)	User Stat. Number	Stat. Label (----)	Repeat Fields 2-3; Maximum = 24														
UTI (48)	User Stat.	Stat. Label (----)	Initial Value (0.)	Repeat Fields 2-4; Maximum = 16													
UHI (49)	User Histo. Number	Histo. Label (----)	Number of Cells (25)	Upper Lim. of Cell 1 (0.)	Cell Width (1.)	Repeat Fields 2-6; Maximum = 9											

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
UPL (51)	User Plot Number	Indep. Axis Label (TIME)	Periph. Storage Unit (0)	Plot Option (P)	Plot Increment (5.)												
UVA (52)	User Plot	Var. Number	Plot Symbol (F3)	Var. Label (----)	Lower Limit Key (M)	Upper Limit Key (M)	Lower Limit Value (0.)	Upper Limit Value (0.)	Repeat Fields 3-9; Maximum = 6								
IMO (54)	Moder. Func. Number	Initial Status (I)	Repeat Fields 2-3; Maximum = 24														
IRA (55)	Resr. Number	Resr. Atrib. Number	Func. Type (SC)	Param. Spec. (0.)	Repeat Fields 3-5; Maximum = 10												
ISA (56)	System Atrib. Number	Func. Type (SC)	Param. Spec. (0.)	Repeat Fields 2-4; Maximum = 10													
LRE (57)	Resr. Number	Resr. Label (----)	Repeat Fields 2-3; Maximum = 10														
LSV (58)	S.V. Number	S.V. Label (----)	Repeat Fields 2-3; Maximum = 10														
TAS (60)	Task Number	Task Label (----)	First Rel. Rqmts. (0/1)	Subs. Rel. Rqmts. (9999)	Func. Type (SC)	Param. Spec. (0)	Resr. Number (0)	Task Prty. (0.)	Spec. Task Char. (NS)	Diff. Pred.? (N)	Info. Choice Mech. (L)	Info. Atrib. Number (0)	Task Comp. Prec. (0.)	Resr. Rqmt. Code (A)	Resr. Number	Repeat Field 16; Maximum = 5	
STA (62)	Task Number	Mark? (N)	Mark Point (COM)	Stat. Type (NO)	Collect Point (COM)	Number of Cells (25)	Upper Lim. of Cell 1 (0.)	Cell Width (1.)									
UTC (63)	Task Number	U.T.C. Value (0.)	Repeat Field 3; Maximum = 25 Field Number=Characteristic Number+2														
MOD (64)	Task Number	Moder. Func. Number	Status Update (D)	Duration of Status (T)	Repeat Fields 3-5; Maximum = 10												

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ATA (65)	Task Number	Assign. Point (C)	Atrib. Type (IA)	Resr. Number (0)	Atrib. Number (0)	Func. Type (&C)	Param. Spec. (0)	Repeat Fields 4-8; Maximum = 6									
DET (67)	Task Number	Succ. Task Number	Repeat Field 3; Maximum = 20														
PRO (68)	Task Number	Atrib. Type (NO)	Resr. Number (0)	Succ. Task Number	Prob. or Atrib.	Repeat Fields 5-6; Maximum = 10											
CFI (69)	Task Number	Succ. Task Number	Con-dition Code (NO)	Value or Atrib. (0.)	Atrib. Number (0)	Atrib. Type (IA/NO)	Resr. Number (0)	Repeat Fields 3-8; Maximum = 5									
CAL (71)	Task Number	Succ. Task Number	Con-dition Code (NO)	Value or Atrib. (0.)	Atrib. Number (0)	Atrib. Type (IA/NO)	Resr. Number (0)	Repeat Fields 3-8; Maximum = 5									
NMO (72)	Task Number	Task to Replace	Task to Insert	Repeat Fields 3-4; Maximum = 10													
DMO (73)	Task Number	Distr. Set to Replace	Distr. Set to Insert	Repeat Fields 3-4; Maximum = 10													
TCL (74)	Task Number	Task to Clear	Task to Signal	Repeat Fields 3-4; Maximum = 10													
RCL (75)	Task Number	Resr. to Clear	Task to Signal	Repeat Fields 3-4; Maximum = 10													
SWI (76)	Task Number	Switch to Change	New Switch Value (0)	Repeat Fields 3-4; Maximum = 10													
REG (77)	Task Number	SS(.) to Reg.	S.V. in Reg. Func. (0)	Mult. Const. (1.)	Add. Const. (0.)	Effect of Reg. (E)	Repeat Fields 3-7; Maximum = 2										

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
MON (79)	Monitor Number	Monitor Label (----)	S.V. to Monitor	S.V. in Thres. Func. (0)	Mult. Const. (1.)	Add. Const. (0.)	Monitor Direct. (B)	Monitor Tol. (0.)									
MTA (80)	Monitor Number	Task to Signal	Repeat Field 3; Maximum = 10														
MSW (81)	Monitor Number	Switch to Change	New Switch Value (0)	Repeat Fields 3-4; Maximum = 10													
SST (82)	S.V. Stat. Number	S.V. to Collect	Stat. Label	Initial Value	Repeat Fields 3-5; Maximum = 10												
PLO (83)	S.V. Plot Number	Indep. Axis Label (TIME)	Periph. Storage Unit (0)	Plot Option (P)	Plot Increment (5.)												
VAR (84)	S.V. Plot Number	Var. Number	S.V. to Plot	Plot Symbol (F3)	Var. Label (----)	Lower Limit Key (M)	Upper Limit Key (M)	Lower Limit Value (0.)	Upper Limit Value (0.)								

SECTION IV

SAINT USER-WRITTEN SUBPROGRAMS

SAINT includes nine user-written subprograms that are used to define additional model components. These subprograms and the functions they can perform are shown in Table 7. By including the appropriate SAINT COMMON blocks, all SAINT COMMON variables are available to the user. In addition, the user has access to the user support subprograms that are described in Section V.

TABLE 7

SAINT USER-WRITTEN SUBPROGRAMS

Subprograms	Function
Subroutine ENDIT(NITER)	Prepares user-generated iteration summary report for iteration NITER and initializes user-generated statistical storage arrays. Automatically called by SAINT at the end of iteration NITER.
Subroutine INTLC	Defines the initial values of the state variable vector SS(·) and the switch vector IS(·). Automatically called by SAINT at the end of state variable data input processing.
Subroutine MODRF(MFN,NNODE)	Computes and causes the effect of moderator function MFN on the performance of task NNODE. The effect can alter TTIME, the baseline task performance, or PFIRB, the probability of taking the first branch specified upon completion of the task. Automatically called by SAINT if moderator function MFN is active when the performance of NNODE is being scheduled.
Function PRIOR(ITASK)	Computes the priority PRIOR of task ITASK prior to the assignment of resources. Automatically called by SAINT immediately preceding the assignment of resources to tasks that have been released.
Subroutine STATE	Contains algebraic, difference, and differential equations which define the state variables of the model. Automatically called by SAINT when the values of state variables are to be updated.

AD-A058 724 PRITSKER AND ASSOCIATES INC WEST LAFAYETTE IND F/G 9/2
THE SAINT USER'S MANUAL.(U)
JUN 78 D B WORTMAN, S D DUKET, D J SEIFERT F33615-76-C-5012
UNCLASSIFIED AMRL-TR-77-62 NL

2 OF 2
AD
A058724



END
DATE
FILMED
11-78
DDC

Subprograms	Function
Subroutine UERR(KODE)	Prepares user-generated error report when a SAINT execution error occurs. Automatically called by SAINT when execution error KODE is detected.
Subroutine UINPT	Reads user-generated input data. Automatically called by SAINT following the processing of all SAINT input data.
Function USERF(JJ)	Determines the value of USERF when the user-written function (UF) is specified on input. The value of JJ is the parameter specification associated with the function.
Subroutine UOTPT	Prepares user-generated simulation summary report for user-generated statistics and other user-defined quantities. Automatically called by SAINT at the end of all required iterations.

SECTION V

SAINT USER SUPPORT SUBPROGRAMS

With the advent of SAINT user-written subprograms, the SAINT user may require direct access to SAINT variables. Since SAINT maintains lists and pointer systems, a number of subprograms are provided to facilitate the retrieval of variable values. These user-callable support subprograms allow the user to access frequently required SAINT variables as well as to collect statistics on any user-defined variables. They are organized by function in the following six tables, where their descriptions and argument definitions are given.

Table Number	Function
8	Attribute Values
9	User-Defined Task Characteristics
10	Task Scheduling
11	Random Deviate Generators
12	User-Generated Statistics
13	Simulation Status

TABLE 8

SAINT USER SUPPORT SUBPROGRAMS:
ATTRIBUTE VALUES

Subprogram	Function
Subroutine GETIA(NAT,VALUE)	Retrieves VALUE as the value of information attribute NAT of the information packet associated with the task now being processed.
Subroutine PUTIA(NAT,VALUE)	Sets information attribute NAT of the information packet associated with the task now being processed to the value of VALUE.
Subroutine GETRA(NRE,NAT,VALUE)	Retrieves VALUE as the value of resource attribute NAT of resource NRE.
Subroutine PUTRA(NRE,NAT,VALUE)	Sets resource attribute NAT of resource NRE to the value of VALUE.
Subroutine GETSA(NAT,VALUE)	Retrieves VALUE as the value of system attribute NAT.
Subroutine PUTSA(NAT,VALUE)	Sets system attribute NAT to the value of VALUE.

TABLE 9

SAINT USER SUPPORT SUBPROGRAMS:
USER-DEFINED TASK CHARACTERISTICS

Subprogram	Function
Subroutine GETTC (NNODE,NCHAR,VALUE)	Retrieves VALUE as the value of user-defined task characteristic NCHAR of task NNODE.
Subroutine PUTTC (NNODE,NCHAR,VALUE)	Sets User-defined task characteristic NCHAR of task NNODE to the value of VALUE.

TABLE 10

SAINT USER SUPPORT SUBPROGRAMS:
TASK SCHEDULING

Subprogram	Function
Subroutine GETPR(ITASK,VALUE)	Retrieves VALUE as the value of the task priority of task ITASK.
Subroutine PUTPR(ITASK,VALUE)	Sets the task priority of task ITASK to the value of VALUE.
Subroutine QRANK(IRANK)	Changes the ranking procedure used in ranking those tasks awaiting the assignment of resources to IRANK, where IRANK is 1 (low value first based on priority), 2 (high value first based on priority), 3 (FIFO), or 4 (LIFO).
Function TIMEQ(IDUM)	Retrieves the time that the task now being processed has been residing in the file of tasks awaiting the assignment of resources. This function is only accessible from function PRIOR. The argument IDUM is a dummy argument required by ANSI FORTRAN.
Function TMARK(IDUM)	Retrieves the mark time contained in the information packet associated with the task now being processed. The argument IDUM is a dummy argument required by ANSI FORTRAN.

TABLE 11

SAINT USER SUPPORT SUBPROGRAMS:
RANDOM DEVIATE GENERATORS

Subprogram	Function
Function BETA(J)	Generates a random deviate from a Beta or Beta-Pert distribution with parameters in distribution set J.
Function DRAND(IY)	Generates a uniform random number DRAND over the interval (0,1) using the computer system's random number routine. This function is machine specific. The argument IY is the initial random number seed (ISEED).
Function ERLNG(J)	Generates a random deviate from an Erlang distribution with parameters in distribution set J.
Function GAMM(J)	Generates a random deviate from a Gamma distribution with parameters in distribution set J.
Subroutine NPOSN(J,NPSSN)	Generates a random deviate NPSSN from a Poisson distribution with parameters in distribution set J.
Function RLOGN(J)	Generates a random deviate from a log-normal distribution with parameters in distribution set J.
Function RNORM(J)	Generates a random deviate from a normal distribution with parameters in distribution set J.
Function TRNGL(J)	Generates a deviate from a triangular distribution with parameters in distribution set J.
Function UNFRM(J)	Generates a random deviate from a uniform distribution with parameters in distribution set J.
Function WEIBL(J)	Generates a random deviate from a Weibull distribution with parameters in distribution set J.

TABLE 12

SAINT USER SUPPORT SUBPROGRAMS:
USER-GENERATED STATISTICS

Subprogram	Function
Subroutine CLEAR	Initializes all statistical storage arrays for user-generated statistics. Automatically called by SAINT following the processing of all input data cards.
Subroutine CLRHI(IND)	Initializes the statistical storage arrays associated with the collection of user-generated histograms. If IND is equal to 0, statistical storage for all histgrams is initialized; if IND is greater than 0, statistical storage for the histogram identified by IND is initialized.
Subroutine CLROB(IND)	Initializes the statistical storage arrays associated with the collection of user-generated statistics for variables based on observation. If IND is equal to 0, statistical storage for all variables based on observation is initialized; if IND is greater than 0, statistical storage for the variable identified by IND is initialized.
Subroutine CLRPT(IND)	Initializes storage arrays and peripheral storage units associated with the preparation of user-generated plots. If IND is equal to 0, storage for all user-generated plots is initialized; if IND is greater than 0, storage for the plot identified by IND is initialized.

Subprogram	Function
Subroutine CLRTP(IND)	Initializes the statistical storage arrays associated with the collection of user-generated statistics for time-persistent variables. If IND is equal to 0, statistical storage for all time-persistent variables is initialized; if IND is greater than 0, statistical storage for the variable identified by IND is initialized.
Subroutine UCLCT(XX,ICLCT)	If ICLCT is greater than 0, records value XX as an observation on variable ICLCT. If ICLCT equals 0, computes and reports statistics on all variables based on observation. If ICLCT is less than 0, computes and reports statistics on variable -ICLCT.
Subroutine UHIST(XX,IHIST)	If IHIST is greater than 0, determines the cell number associated with the value XX for histogram IHIST and increases the cell content by 1. If IHIST equals 0, prints and plots all user-generated histograms. If IHIST is less than 0, prints and plots user-generated histogram -IHIST.
Subroutine UPLOT(XX,T,IPLOT)	If IPLOT is greater than 0, IPLOT is the plot number and UPLOT stores values of the dependent variables XX (up to ten values per plot) for a value of the independent variable T. If IPLOT equals 0, prints table and/or plot for all user-generated plots. If IPLOT is less than 0, prints table and/or plot for user-generated plot -IPLOT.

Subprogram	Function
Subroutine UTMSA(XX,T,ISTAT)	Integrates variable ISTAT assuming that the value during the past interval is the average of XX and the value XX at the last cell to UTMSA for variable ISTAT. Computation and reporting for variable ISTAT is performed by subroutine UTMST.
Subroutine UTMST(XX,T,ISTAT)	If ISTAT is greater than 0, integrates variable ISTAT assuming that the value during the interval up to time T is the value of XX at the last call to UTMST for variable ISTAT. XX is the value that variable ISTAT will have during the next interval. If ISTAT equals 0, computes and reports statistics on all user-generated time-persistent variables. If ISTAT is less than 0, computes and reports statistics on user-generated time-persistent variable -ISTAT.

TABLE 13

SAINT USER SUPPORT SUBPROGRAMS:
SIMULATION STATUS

Subprogram	Function
Subroutine DMPAR	Prints the current status of all tasks awaiting the assignment of resources.
Subroutine DMPMF	Prints the current status of all moderator functions.
Subroutine DMPRS	Prints the current status of all resources.
Subroutine DMPSA	Prints the current status of all system attributes.
Subroutine DMPST	Prints the current status of all tasks now being performed.
Subroutine DMPSV	Prints the current status of all state variables.

SECTION VI

SAINT INPUT ERRORS

Two separate and self-contained input processors are available for use with the SAINT simulation program. The first processor performs maximum error checking. This processor outputs primary error specifications following the individual card images on the echo check, and secondary error specifications following the complete card image echo check. When a SAINT input error occurs, the error will be identified by a printed error message describing the error and its location.

The second input processor performs no error checking. It should be used only when the SAINT user is certain that no errors appear when using the maximum error checking input processor.

With these capabilities, the user will be able to initially check his input data using the maximum error checking processor and then make subsequent runs with the same data deck using the no error checking processor in order to save computer time. For both input processors, all undefined input variables, except those required by SAINT, will be given default values.

SECTION VII

SAINT EXECUTION ERRORS

The SAINT simulation program is designed to monitor its own execution and detect any errors that arise. When an error is detected, execution is automatically terminated by calculating the square root of a negative number, so that subprogram traceback information will be printed. SAINT then provides complete information detailing the status of the system at the point in time when the error occurs. This information includes the error code number, the time that the error occurred, the iteration being processed, as well as the current status of all tasks, resources, attributes, moderator functions, and state variables.

When a SAINT execution error occurs, the error will be identified by an error code number. Table 14 provides the description, probable cause, and location associated with each error code. The information included in Table 14 along with the status report printed when the error occurs should provide sufficient information to locate and correct the error.

Following the printing of the error report, SAINT automatically calls subroutine UERR(KODE). This subroutine is user-written and may be used to generate user-required error reports on the status of user-defined variables. The argument KODE is set equal to the code number of the error detected.

TABLE 14

SAINT EXECUTION ERROR DESCRIPTIONS

Error Code	Error Condition	Location of Call to SERR	
3001	Failed integration accuracy test; user specified execution abort. Probable cause: specified accuracy requirements are too strict.	UPDATE	144
3002	Failed accuracy test in locating threshold crossing; user specified execution abort. Probable cause: specified accuracy requirements are too strict.	UPDATE	189
3003	Sum of the branching probabilities for a task that requires probabilistic branching does not equal 1.0. Probable cause: incorrect assignment of branching probabilities during execution.	GASP	372
3004	One of the branching probabilities for a task that requires probabilistic branching is less than 0. Probable cause: incorrect assignment of branching probability during execution.	GASP	369
3005	All available storage for information attribute packets has been allocated. Probable cause: associated array dimensions are too small (see Table 16, No. 13, for arrays affected).	GASP	478
3006	All available storage for tasks being performed and tasks awaiting assignment of resources has been allocated. Probable cause: associated array dimensions are too small (see Table 16, No. 1, for arrays affected).	FILEM	11
3007	Value of the variable TTIME (task performance time) is less than 0. Probable cause: incorrect specification of variable value in user-written subprogram.	SCHAT	67

Error Code	Error Condition	Location of Call to SERR	
3008	Value of the variable PFIRB (probability of selecting the first branch for a task that requires probabilistic branching) is less than 0. Probable cause: incorrect specification of variable value in user-written subprogram.	SCHAT	74
3009	Value of the variable PFIRB (probability of selecting the first branch for a task that requires probabilistic branching) is greater than 1. Probable cause: incorrect specification of variable value in user-written subprogram.	SCHAT	75
3010	Not all sink tasks required for completion of simulation have been performed. However, no other tasks are being performed. Probable cause: incorrect branching specifications (incorrect network formulation).	RMOVE	14
3101	Distribution set number specified as an argument to this subprogram refers to a distribution set that contains an illegal distribution type for this subprogram. Probable cause: incorrect specification of argument in calling subprogram.	BETA	7
3102	Distribution set number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	BETA	4
3103	Distribution set number specified as an argument to this subprogram is greater than the largest distribution set number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	BETA	5
3104	Distribution set number specified as an argument to this subprogram refers to a distribution set that contains an illegal distribution type for this subprogram. Probable cause: incorrect specification of argument in calling subprogram.	ERLNG	8

Error Code	Error Condition	Location of Call to SERR	
3105	Distribution set number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	ERLNG	5
3106	Distribution set number specified as an argument to this subprogram is greater than the largest distribution set number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	ERLNG	6
3107	Distribution set number specified as an argument to this subprogram refers to a distribution set that contains an illegal distribution type for this subprogram. Probable cause: incorrect specification of argument in calling subprogram.	GAMM	7
3108	Distribution set number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	GAMM	4
3109	Distribution set number specified as an argument to this subprogram is greater than the largest distribution set number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	GAMM	5
3110	Distribution set number specified as an argument to this subprogram refers to a distribution set that contains an illegal distribution type for this subprogram. Probable cause: incorrect specification of argument in calling subprogram.	NPOSN	9
3111	Distribution set number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	NPOSN	6

Error Code	Error Condition	Location of Call to SERR	
3112	Distribution set number specified as an argument to this subprogram is greater than the largest distribution set number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	NPOSN	7
3113	Distribution set number specified as an argument to this subprogram refers to a distribution set that contains an illegal distribution type for this subprogram. Probable cause: incorrect specification of argument in calling subprogram.	RLOGN	7
3114	Distribution set number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	RLOGN	4
3115	Distribution set number specified as an argument to this subprogram is greater than the largest distribution set number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	RLOGN	5
3116	Distribution set number specified as an argument to this subprogram refers to a distribution set that contains an illegal distribution type for this subprogram. Probable cause: incorrect specification of argument in calling subprogram.	RNORM	8
3117	Distribution set number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	RNORM	5
3118	Distribution set number specified as an argument to this subprogram is greater than the largest distribution set number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	RNORM	6

Error Code	Error Condition	Location of Call to SERR	
3119	Distribution set number specified as an argument to this subprogram refers to a distribution set that contains an illegal distribution type for this subprogram. Probable cause: incorrect specification of argument in calling subprogram.	TRNGL	8
3120	Distribution set number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	TRNGL	6
3121	Distribution set number specified as an argument to this subprogram is greater than the largest distribution set number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	TRNGL	7
3122	Distribution set number specified as an argument to this subprogram refers to a distribution set that contains an illegal distribution type for this subprogram. Probable cause: incorrect specification of argument in calling subprogram.	UNFRM	8
3123	Distribution set number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	UNFRM	5
3124	Distribution set number specified as an argument to this subprogram is greater than the largest distribution set number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	UNFRM	6
3125	Distribution set number specified as an argument to this subprogram refers to a distribution set that contains an illegal distribution type for this subprogram. Probable cause: Incorrect specification of argument in calling subprogram.	WEIBL	8

Error Code	Error Condition	Location of Call to SERR	
3126	Distribution set number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	WEIBL	5
3127	Distribution set number specified as an argument to this subprogram is greater than the largest distribution set number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	WEIBL	6
3201	Histogram number specified as an argument to this subprogram is less than 0. Probable cause: incorrect specification of argument in calling subprogram.	CLRHI	5
3202	Histogram number specified as an argument to this subprogram is greater than the largest histogram number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	CLRHI	6
3203	Statistic number specified as an argument to this subprogram is less than 0. Probable cause: incorrect specification of argument in calling subprogram.	CLROB	5
3204	Statistic number specified as an argument to this subprogram is greater than the largest statistic number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	CLROB	6
3205	Plot number specified as an argument to this subprogram is less than 0. Probable cause: incorrect specification of argument in calling subprogram.	CLRPT	5
3206	Plot number specified as an argument to this subprogram is greater than the largest plot number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	CLRPT	6
3207	Statistic number specified as an argument to this subprogram is less than 0. Probable cause: incorrect specification of argument in calling subprogram.	CLRTP	6

Error Code	Error Condition	Location of Call to SERR	
3208	Statistic number specified as an argument to this subprogram is greater than the largest statistic number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	CLRTP	7
3209	Clearing of user-generated histograms is requested. However, no user-generated histograms are defined. Probable cause: call made to inappropriate clearing subprogram from a user-written subprogram.	CLRHI	4
3210	Clearing of user-generated statistics based on observation is requested. However, no user-generated statistics based on observation are defined. Probable cause: call made to inappropriate clearing subprogram from a user-written subprogram.	CLROB	4
3211	Clearing of user-generated statistics for time persistent variables is requested. However, no user-generated statistics for time persistent variables are defined. Probable cause: call made to inappropriate clearing subprogram from a user-written subprogram.	CLRTP	5
3212	Clearing of user-generated plots is requested. However, no user-generated plots are defined. Probable cause: call made to inappropriate clearing subprogram from a user-written subprogram.	CLRPT	4
3301	Statistic number specified as an argument to this subprogram for output purposes is greater than the largest statistic number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	UCLCT	9
3302	Statistic number specified as an argument to this subprogram is greater than the largest statistic number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	UCLCT	16

Error Code	Error Condition	Location of Call to SERR	
3303	Histogram number specified as an argument to this subprogram for output purposes is greater than the largest histogram number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	UHIST	10
3304	Histogram number specified as an argument to this subprogram is greater than the largest histogram number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	UHIST	20
3305	Statistic number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	UTMSA	6
3306	Statistic number specified as an argument to this subprogram is greater than the largest statistic number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	UTMSA	7
3307	Value of the independent variable specified as an argument to this subprogram is not greater than or equal to the last specified value of the independent variable. Probable cause: incorrect specification of argument in calling subprogram.	UTMSA	11
3308	Statistic number specified as an argument to this subprogram for output purposes is greater than the largest statistic number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	UTMST	9
3309	Statistic number specified as an argument to this subprogram is greater than the largest statistic number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	UTMST	16

Error Code	Error Condition	Location of Call to SERR	
3310	Value of the independent variable specified as an argument to this subprogram is not greater than or equal to the last specified value of the independent variable. Probable cause: incorrect specification of argument in calling subprogram.	UTMST	18
3311	Plot number specified as an argument to this subprogram for output purposes is greater than the largest plot number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	UPLOT	29
3312	Plot number specified as an argument to this subprogram is greater than the largest plot number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	UPLOT	11
3313	Value of the independent variable specified as an argument to this subprogram is not monotonic. Probable cause: incorrect specification of argument in calling subprogram.	UPLOT	132
3314	Collection or reporting of user-generated histograms is requested. However, no user-generated histograms are defined. Probable cause: call made to inappropriate subprogram from a user-written subprogram.	UHIST	7
3315	Collection or reporting of user-generated statistics based on observation is requested. However, no user-generated statistics based on observation are defined. Probable cause: call made to inappropriate subprogram from a user-written subprogram.	UCLCT	6
3316	Collection or reporting of user-generated statistics for time persistent variables is requested. However, no user-generated statistics for time persistent variables are defined. Probable cause: call made to inappropriate subprogram from a user-written subprogram.	UTMSA	5

Error Code	Error Condition	Location of Call to SERR	
3317	Collection or reporting of user-generated statistics for time persistent variables is requested. However, no user-generated statistics for time persistent variables are defined. Probable cause: call made to inappropriate subprogram from a user-written subprogram.	UTMST	6
3318	Collection or reporting of user-generated plots is requested. However, no user-generated plots are defined. Probable cause: call made to inappropriate subprogram from a user-written subprogram.	UPLOT	9
3319	The number of variables for the plot specified as an argument to this subprogram is 0. Collection for this plot has been requested. Probable cause: incorrect specification of argument in calling subprogram.	UPLOT	14
3320	The number of variables for the plot specified as an argument to this subprogram is 0. Reporting for this plot has been requested. Probable cause: incorrect specification of argument in calling subprogram.	UPLOT	30
3401	Attribute number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	GETIA	5
3402	Attribute number specified as an argument to this subprogram is greater than the largest attribute number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	GETIA	6
3403	Task number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	GETPR	4
3404	Task number specified as an argument to this subprogram is greater than the largest task number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	GETPR	5

Error Code	Error Condition	Location of Call to SERR	
3405	Attribute number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification in calling subprogram.	GETRA	5
3406	Attribute number specified as an argument to this subprogram is greater than the largest attribute number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	GETRA	6
3407	Resource number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification in argument in calling subprogram.	GETRA	7
3408	Resource number specified as an argument to this subprogram is greater than the largest resource number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	GETRA	8
3409	Attribute number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	GETSA	4
3410	Attribute number specified as an argument to this subprogram is greater than the largest attribute number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	GETSA	5
3411	Characteristic number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	GETTC	7
3412	Characteristic number specified as an argument to this subprogram is not defined for the task specified as an argument. Probable cause: incorrect specification of argument in calling subprogram.	GETTC	8

Error Code	Error Condition	Location of Call to SERR	
3413	Task number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	GETTC	5
3414	Task number specified as an argument to this subprogram is greater than the largest task number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	GETTC	6
3501	Attribute number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	PUTIA	5
3502	Attribute number specified as an argument to this subprogram is greater than the largest attribute number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	PUTIA	6
3503	Task number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	PUTPR	4
3504	Task number specified as an argument to this subprogram is greater than the largest task number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	PUTPR	5
3505	Attribute number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification in calling subprogram.	PUTRA	7
3506	Attribute number specified as an argument to this subprogram is greater than the largest attribute number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	PUTRA	8
3507	Resource number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	PUTRA	5

Error Code	Error Condition	Location of Call to SERR	
3508	Resource number specified as an argument to this subprogram is greater than the largest resource number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	PUTRA	6
3509	Attribute number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	PUTSA	4
3510	Attribute number specified as an argument to this subprogram is greater than the largest attribute number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	PUTSA	5
3511	Characteristic number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	PUTTC	7
3512	Characteristic number specified as an argument to this subprogram is not defined for the task specified as an argument. Probable cause: incorrect specification of argument in calling subprogram.	PUTTC	8
3513	Task number specified as an argument to this subprogram is less than or equal to 0. Probable cause: incorrect specification of argument in calling subprogram.	PUTTC	5
3514	Task number specified as an argument to this subprogram is greater than the largest task number specified on input. Probable cause: incorrect specification of argument in calling subprogram.	PUTTC	6
3515	Ranking rule specified as an argument to this subprogram is not equal to 1, 2, 3, or 4. Probable cause: incorrect specification of argument in calling subprogram.	QRANK	5
3601	Report on the status of moderator functions is requested. However, no moderator functions are defined. Probable cause: call made to inappropriate status reporting subprogram from a user-written subprogram.	DMPMF	10

Error Code	Error Condition	Location of Call to SERR	
3602	Report on status of resources is requested. However, no resources are defined. Probable cause: call made to inappropriate status reporting subprogram from a user-written subprogram.	DMPRS	14
3603	Report on status of system attributes is requested. However, no system attributes are defined. Probable cause: call made to inappropriate status reporting subprogram from a user-written subprogram.	DMPSA	10
3604	Report on status of state variables is requested. However, no state variables are defined. Probable cause: call made to inappropriate status reporting subprogram from a user-written subprogram.	DMPSV	8

SECTION VIII

SAINT OPERATING PROCEDURES

SAINT is written in ANSI standard FORTRAN IV and can be run on any machine that has a FORTRAN compiler. In this and subsequent sections, general information relative to the operation of the SAINT program is provided.

SAINT normally utilizes four input and output devices. The variable NCRDR is set to the input device number. The variable NPRNT is set to the output device number. The variable NRNIT is set to a peripheral device number to be used during input processing and initialization prior to each iteration. Output to this device is formatted. The variable NRENT is set to a peripheral device number to be used during the reinitialization of program variables at the end of each iteration. The values of these variables are specified in the main program. In the version of SAINT documented herein and available for distribution, these variables are defined as NCRDR=5, NPRNT=6, NRNIT=8, and NRENT=9. The variable NPUNCH is also defined in the main program for the convenience of the user in generating punched card output. It is defined as NPUNCH=7. In addition, SAINT utilizes peripheral devices in the preparation of state variable and user-generated plots. The device numbers to be used are specified by the user on SAINT data cards. The devices should be made available through the use of proper JCL for the user's machine.

In SAINT, function DRAND is the only machine-specific subprogram. Function DRAND is used by SAINT to generate a pseudo-random number in the interval (0,1). It is suggested that the coding of function DRAND be replaced by the random number generator in use at your computer center.

SECTION IX

SAINT OVERLAY STRUCTURE

The basic philosophy employed in the design of SAINT was that additions, modifications, and deletions are to be expected. Because of this, the SAINT computer program was designed in a modular form to facilitate adaptation. This same design philosophy allows efficient use of core storage by providing the means to develop a storage-saving overlay structure.

In designing the SAINT overlay structure, the SAINT subprograms were divided into four groups. The first group of subprograms remains in core storage throughout the simulation process. The other three groups are brought into core storage one at a time, dependent upon the status of the simulation being performed. Table 15 shows the manner in which the SAINT subprograms were divided into groups.

Group 1 remains in core storage throughout the program's execution, as the subprograms contained within Group 1 are accessed from more than one of the remaining three subprogram groups. When the main program calls subroutine DATIN to begin processing the SAINT data cards, group 2 is brought into core storage along with group 1.

After processing all data cards, subroutine DATIN returns control to the main program. The main program then calls subroutine GASP, causing group 2 to be removed from core storage and group 3 to be brought in. Group 3 contains all subprograms necessary to perform the simulation.

When all iterations have been completed, subroutine GASP returns control to the main program. The main program then calls subroutine SUMRY for the purpose of preparing the required output reports. This causes group 3 to be removed from core storage and group 4 to be inserted.

Following the completion of the summary reports, subroutine SUMRY returns control to the main program. If no additional SAINT networks are to be simulated, execution is terminated. However, if additional networks are to be analyzed, the entire simulation process is restarted with a call to DATIN from the main program.

TABLE 15

GROUPINGS OF SAINT SUBPROGRAMS

Group 1	Group 2		Group 3		Group 4
ASSGN	ATASS	LPACK	ATSET	QRANK	SUMRY
BETA	BETAXF	MSWT	COLST	RMOVE	UOTPT
CLEAR	BUILD	MTASK	ENDIT	RPLOT	
CLRHI	CNCVT	NMOD	FILEM	SCHAT	
CLROB	CONDIT	OUTPT	GASP	SCHED	
CLRPT	CVT	PERTXF	GETIA	SCOND	
CLRTP	DATIN	PLOTS	GETPR	SSAVE	
DMPAR	DET	PNABA	GETRA	TIMEQ	
DMPFL	DFAUS	POP	GETSA	TMARK	
DMPMF	DFAUT	PROB	GETTC	UPDATE	
DMPRS	DISTR	RCLEAR	MODRF		
DMPSA	DMODS	REG	NFIND		
DMPST	DSWT	SGEN	PRIOR		
DMPSV	ECHO	STAT	PUTIA		
DRAND	ECHOS	STATT	PUTPR		
ERLNG	ERRIN	TASK	PUTRA		
GAM	GEN	TCLEAR	PUTSA		
GAMM	GTCHAR	UCOLL	PUTTC		
HISTO	IMODFN	UHSTO			
INTLC	INIT	UINPT			
NPOSN	INITS	UPLTS			
PRNTH	IRATT	UTIME			
PRNTS	ISATT	UVAR			
RLOGN	MAP	VAR			
RNORM	MODFN				
SERR	MONIT				
STATE	LLABL				
TRNGL					
UCLCT					
UERR					
UHIST					
UNFRM					
UPLOT					
USERF					
UTMSA					
UTMST					
WEIBL					

SECTION X

DIMENSION REQUIREMENTS FOR SAINT ARRAYS

The version of the SAINT simulation program documented herein and available for distribution has been designed to accommodate a model with the characteristics listed in Table 16.

The model characteristics that the SAINT program will accept can be altered by changing the values of the variables given in column 3 of Table 16 to the desired values in the main program and by changing the dimensions of the arrays affected (column 5) in the main program and all SAINT subprograms where their associated COMMON block appears. The main program for SAINT is shown in Figure 7. A general description of the characteristic associated with each COMMON block appears in Table 17. The COMMON blocks included in each SAINT subprogram are shown in Table 18. The definitions of all SAINT COMMON variables are provided in Documentation for the SAINT Simulation Program (2).

For example, assume that a SAINT model requires that the largest task number be 500. Entry 3 of Table 16 provides the information for altering the largest task number. The name of the variable that is set to the largest task number is IMN (presently set to 100). Thus, the first step in the alteration process is to remove the following statement from the main program:

```
IMN = 100
```

and replace it with the statement:

```
IMN = 500.
```

Once this has been accomplished, the next step in the alteration process involves changing the dimensions of the arrays affected, as shown in column 5, from 100 to 500. Thus, all dimensioned arrays which appear in the COMMON block COM10 must be altered. The arrays included in COM10 are shown in the main program (Figure 7). The new version of COM10, where all arrays are dimensioned to 500, will replace the old version in the main program. In addition, Table 18 indicates that COM10 must also be replaced in 32 SAINT subprograms.

As soon as the alterations described above have been completed, the SAINT program will accept models whose largest task number is 500. The same alteration procedure is to be followed in order to change any of the other model characteristics.

TABLE 16

DIMENSION REQUIREMENTS FOR SAINT ARRAYS

Controlling Characteristic	Current Maximum Value	Variable Name for Maximum Value	Labeled COMMON Block Containing Arrays Affected	Arrays Affected
1. Number of tasks being performed + number of tasks awaiting assignment of resources.	100	ID	COM02	NSET(ID*4), QSET(ID*3)
2. Largest distribution set number.	100	MPARM	COM09	All dimensioned arrays in COM09
3. Largest task number.	100	IMN	COM10	All dimensioned arrays in COM10
4. Largest resource number.	20	MOPNO	COM11	BUSY, LLRES,NBUS,NOPTR,TLST, RSTAT(MOPNO*4)
5. Number of tasks that require resources + total number of task-resource associations specified.	600	MNOPA	COM11	NOPA
6. (Number of tasks that have branches emanating from their output side * 2) + number of tasks which require probabilistic branching + total number of deterministic branches specified + (total number of probabilistic branches specified * 2) + (total number of conditional branches specified * 5).	850	MVABA	COM12	VABA
7. Number of tasks that cause task modifications + (total number of task modifications specified * 2) + number of tasks that cause distribution set modifications + (total number of distribution set modifications specified * 2) + number of tasks that cause task clearing + (total number of task clearings specified * 2) + number of tasks that cause resource clearing + (total number of resource clearings * 2) + number of tasks that require different predecessors + total number of predecessors to tasks that require different predecessors.	250	IMNA	COM12	NABA

Controlling Characteristic	Current Maximum Value	Variable Name for Maximum Value	Labeled COMMON Block Containing Arrays Affected	Arrays Affected
8. Number of user-defined task characteristics for all tasks (the number of user-defined task characteristics for a task is equal to the largest characteristic number defined for that task).	200	MTCHR	COM12	STCHR
9. Largest moderator function number.	20	MMDFN	COM13	MDFNS,MFSTW
10. Number of tasks at which moderator function status is updated + (total number of moderator function status updates specified * 2).	300	MMSTU	COM13	MFSTU
11. Number of statistics tasks.	50	MXSTA	COM14	NSINK,KSTPE,KSTTM,XSTUS,NCELS, XLOW, WIDTH, SUMAI, SUMAF
12. (Number of statistics tasks * 2) + total number of cells specified for statistics task histograms.	1350	MNCLS	COM14	JCELS
13. Number of information packets in the network * (number of attributes per packet + 1).	1000	MAXDS	COM15	DESCR
14. Largest resource number * number of resource attributes per resource.	100	MDOAT	COM15	DOATT
15. (Number of tasks at which attribute assignments are to be made * 2) + (total number of attribute assignments to be made * 4).	800	MDSTR	COM15	NDSTR
16. Largest system attribute number.	100	MSYAT	COM15	SYSAT
17. Largest index for state variable equations (SS(·) or DD(·) variables).	100	MEQT	COM17	All dimensioned arrays in COM17[1]
18. Largest switch number.	20	MNSWA	COM18	IS

Controlling Characteristic	Current Maximum Value	Variable Name for Maximum Value	Labeled COMMON Block Containing Arrays Affected	Arrays Affected
19. Number of tasks at which switch values are changed + (total number of specified switch value changes as a result of task completion * 2).	300	MDAD	COM18	NABAD
20. Number of tasks at which state variable values are regulated + (total number of specified state variable regulations as a result of task completion * 5).	600	MDDR	COM18	YABAR
21. Largest state variable monitor number.	20	MFLAG	COM19	LFLAG,NPOSS,NPOST,LLMON, THRES(MFLAG*6)
22. Number of state variable monitors causing tasks to be signaled + total number of specified task signalings as a result of monitor action.	40	MDNPT	COM19	NABAT
23. Number of state variable monitors causing switch values to be changed + (total number of specified switch value changes as a result of monitor action * 2).	60	MDNSS	COM19	NABAS
24. Number of state variables for which statistics are to be collected.	20	MSTAT	COM20	All dimensioned arrays in COM20
25. Largest state variable plot number.	10	MPLOT	COM21	DTPLT,IITAP,NNPTS,NNVAR,NNVP, NVP(MPLOT*MNVPP)
26. Largest number of variables plotted on any state variable plot.	10	MNVPP	COM21	LLPHI,LLPLO,LLSYM,PPHI,PPLO, NVP(MPLOT*MNVPP),LLSVP(MNVPP + 1,2)
27. Number of state variable core plot point sets * (number of variables being plotted + 1).	1100	MDQP	COM21	QPSET

Controlling Characteristic	Current Maximum Value	Variable Name for Maximum Value	Labeled COMMON Block Containing Arrays Affected	Arrays Affected
28. Largest statistic number for user-generated statistics based on observation.	20	MNCLT	COM23	LLUGC,USOBV
29. Largest statistic number for user-generated statistics based on time-persistent variables.	20	MNSTP	COM23	LLUGT,TTCLR,USTPV
30. Largest user-generated histogram number.	20	MNHIS	COM23	LLUGH,NNCEL,HHLOW,HHWID
31. (Number of user-generated histograms * 2) + total number of cells specified for user-generated histograms.	540	MNCEL	COM23	JJCEL
32. Largest user-generated plot number.	10	MNPLT	COM24	DPLOT,ITAPE,NPTSV,NVARS
33. Largest number of variables plotted on any user-generated plot.	10	MNVAR	COM24	LPHIH,LPLOW,LSYMB,PHIH,PLOW, LLUGP(MNVAR+1,2)
34. Number of user-generated core plot point sets * (number of variables being plotted + 1)	1100	MNCUP	COM24	UPSET

[1]The following arrays appear in a DIMENSION statement in subroutine UPDATE:
SI,DI,A1,A2,A3,A4,A5,A6,A7
The dimensions of these arrays must be changed in accordance with any changes made to the maximum number of state variables allowed.

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      COMMON /COM01/ ID,IM,IMM,IMNA,IMN,MAXDS,MDAD,MDDR,MDNPT,MDNSS,
     *               MDCAT,MDOP,MDSTR,MEQT,MFLAG,MMPTS,MMDFN,MMSTU,
     *               MNCEL,MNCLS,MNCLT,MNCUP,MNHIS,MNDPA,MNPLT,MNPTC,
     *               MNSTP,MNSWA,MNVAP,MNVPP,MOPNO,MPARM,MPLOT,MSTAT,
     *               MSYAT,MTCHR,MXSTA,MYABA,NOO
      COMMON /COM02/ ATRIB(3),JTRIB(2),QSET(300),NSET(400),MFA,MXX,
     *               MFE(3),MLE(3),NQ(3)
      COMMON /COM03/ IPOS,JPOS,KPOS,LPOS,MPOS,NPA,NAN,IERRW,IERRF,IFIN,
     *               IIECH,INDXS,INDXT,INDX,JNDX,KNDX,IP,NUMFL,ICONT,
     *               IISED,HIVAL,IBLNK,IZERO,LA,LB,LC,LD,LE,LF,LG,LH,LI,
     *               LJ,LK,LL,LM,LN,LO,LP,LQ,LR,LS,LT,LU,LV,LW,LX,LY,LZ
      COMMON /COM04/ IDFAL(4),KREAD(40),IFLAG(50),IRSUL(50),RESUL(50),
     *               IAEC(8,50),KARD(80),IDIG(9)
      COMMON /COM05/ NPROJ,MON,NDAY,NYR,NAME(2),NRUN,NRUNS,NSKSR,
     *               NSKST,LLCVD,NNEQD,NNEQS,NNEQT
      COMMON /COM06/ TNOW,TTNEX,MFAD,SEED,ISEED,NCRDR,NPRNT,NPUNCH,
     *               NRNIT,NRENT,NNDC,NCC,NDTN,NNTC
      COMMON /COM07/ NDE,NOPAT,NSYAT,NDOP,NNM,NNPA,NMDFN,NN,NPRMS,IFLPR,
     *               JFLPR,KRNK,XINN,NFLAG,NNCLT,NNHIS,NNPLT,NNSTA,
     *               NNSTP,NPLOT,NSTTS
      COMMON /COM08/ NEIP,NEIS,NSIP,NSIS,ITRACE,JTRACE,NRTSP,NRTEP,
     *               KTRACE,MTRACE,IIPSR,IISSR,IGRAF,JGRAF,IJTRAC,
     *               NSVVS,NSVVE,NTSOE,NTSOS,LTRACE,NRTSS,NRTES
      COMMON /COM09/ PARAM(100,5),NPTBU(100),PARM1(100),PARM4(100)
      COMMON /COM10/ CACIN(100,3),ITCHR(100),LLTSK(100,2),LSINK(100),
     *               MACIN(100,4),MFEN(100),MFSTT(100),NFTBU(100),
     *               MOP(100),MPO(100),NDCH(100,2),NDEL(100),NOPT(100),
     *               NPAR(100),NPO(100),NPOOR(100),NPODS(100),NPOP(100),
     *               NPOR(100),NPSGN(100),NREL(100),NRELP(100),
     *               NREL2(100),NSIGN(100),NTC(100),NTCHR(100),
     *               NTYPE(100),KMARK(100),XMARK(100)
      COMMON /COM11/ BUSY(20),LLRES(20,2),NBUS(20),NOPTR(20),TLST(20),
     *               NOPA(600),RSTAT(80)
      COMMON /COM12/ YABA(850),NABA(250),STCHR(200)
      COMMON /COM13/ MDFNS(20),MFSTW(20),MFSTU(300)
      COMMON /COM14/ NSINK(50),KSTPE(50),KSTTM(50),XSTUS(50),NCELS(50),
     *               XLOW(50),WIDTH(50),SUMAI(50,5),SUMAF(50,5),
     *               JCELS(1350)
      COMMON /COM15/ DESCR(1000),DOATT(100),NDSTR(800),SYSAT(100)
      COMMON /COM16/ AAERR,DTMAX,DTMIN,DTSAV,IITES,LLERR,RRERR,TTLAS,
     *               TTSAV,DTSUG,DTFUL,DTNOW,ISEES,RESLS,DTACC,LLSAV,
     *               LSAVE
      COMMON /COM17/ SS(100),SSL(100),DD(100),DDL(100),LLSVR(100,2)
      COMMON /COM18/ IS(20),NABAD(300),YABAR(600)
      COMMON /COM19/ LFLAG(20),NPOSS(20),NPOST(20),LLMON(20,2),
     *               NABAT(40),NABAS(60),THRES(120)
      COMMON /COM20/ NSTAT(20),LLSVS(20,2),SSTPV(20,6)
      COMMON /COM21/ DTPLT(10),IITAP(10),NNPTS(10),NNVAR(10),NNVP(10),
     *               LLPLT,NNPT,LLPHI(10),LLPLO(10),LLSYM(10),PPHI(10),
     *               PPLO(10),NVP(100),LLSVP(11,2),QPSET(1100)
      COMMON /COM22/ TTIME,PFIRB
      COMMON /COM23/ LLUGC(20,2),USOBV(20,5),LLUGT(20,2),TTCLR(20),
     *               USTPV(20,6),LLUGH(20,2),NNCEL(20),HHLOW(20),
     *               HHWID(20),JJCEL(540)
      COMMON /COM24/ DPLOT(10),ITAPE(10),NPTSV(10),NVARS(10),LPLOT,
     *               NPTEX,LPHIH(10),LPLOW(10),LSYMB(10),PHIM(10),
     *               PLOW(10),LLUGP(11,2),UPSET(1100)
```

Figure 7(1). Main Program for SAINT.

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
C*****INITIALIZE READ, PRINT, AND PUNCH KEYS
C
      NCRDR=5
      NPRNT=6
      NPUNCH=7
      NRNIT=8
      NRENT=9
C
C*****DEFINE VARIABLES WHICH REPRESENT ARRAY MAXIMA AND SIMULATION
C*****LIMITS
C
      ID=100
      IMN=100
      IMNA=250
      MAXCS=1000
      MCAD=300
      MDDR=600
      MDNPT=40
      MDNSS=60
      MDOAT=100
      MDOP=1100
      MDSTR=800
      MEQT=100
      MFLAG=20
      MMOFN=20
      MMSTU=300
      MNCEL=540
      MNCLS=1350
      MNCLT=20
      MNCUP=1100
      MNHIS=20
      MNOPA=600
      MNPLT=10
      MNSTP=20
      MNSWA=20
      MNVAR=10
      MNVPP=10
      MCPNO=20
      MPARM=100
      MPLOT=10
      MSTAT=20
      MSYAT=100
      MTCHR=200
      MXSTA=50
      MYABA=950
C
C*****EXECUTION CYCLE -- READ INPUT DATA, THEN INITIATE THE SIMULATION
C
      IFIN=0
  100 CALL DATIN
      IF(IERRF.EQ.0) CALL GASP
      IF(IERRF.EQ.0) CALL SUMRY
      IF (IFIN.EQ.0) GO TO 100
      STOP
C
      END
```

Figure 7(2). Main Program for SAINT.

TABLE 17

SAINT COMMON BLOCK CHARACTERISTICS

COMMON Block	Characteristics
COM01	Maximum Array Dimensions
COM02	Filing System
COM03	Input Processing I
COM04	Input Processing II
COM05	General Information
COM06	Simulation Control
COM07	Program Options
COM08	Output Options
COM09	Distribution Sets
COM10	Tasks I
COM11	Resources
COM12	Tasks II
COM13	Moderator Functions I
COM14	Statistics Tasks
COM15	Attributes
COM16	State Variable Updating
COM17	State Variables
COM18	Switching and Regulation
COM19	State Variable Monitors
COM20	State Variable Statistics
COM21	State Variable Plots
COM22	Moderator Functions II
COM23	User-Generated Statistics
COM24	User-Generated Plots

TABLE 18

COMMON BLOCKS INCLUDED IN SAINT SUBPROGRAMS

Subprogram Name	COMMON Block Number																								
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Main Program	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
ASSGN		x				x			x	x	x				x							x		
ATASS	x		x	x			x		x	x					x									
ATSET		x								x	x				x							x		
BETA							x		x															
BETAXF									x															
BUILD			x	x																				
CLEAR							x																	
CLRHI							x																x	
CLROB							x																x	
CLPRT							x																	x
CLRTP						x	x																x	
CNCVT			x	x																				
COLST					x	x	x	x						x	x									
CONDIT	x		x	x			x			x		x												
CVT			x	x																				
DATIN	x		x	x	x	x	x																	
DET	x		x	x						x		x												
DFAUS	x		x	x	x		x									x	x	x	x	x	x			
DFAUT	x		x	x	x	x	x	x	x	x	x	x	x	x	x								x	x
DISTR	x			x			x		x															
DMODS	x		x	x			x		x	x														
DMPAR		x			x	x																		
DMPFL	x	x				x	x			x					x									
DMPMF					x	x	x						x											
DMPRS					x	x	x				x				x									
DMPSA					x	x	x								x									
DMPST		x			x	x																		
DMPSV					x																			

Subprogram Name	COMMON Block Number																							
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DSWT	x		x	x						x								x					
ECHO	x		x	x	x	x	x	x	x	x	x	x	x	x	x								x	x
ECHOS	x		x	x	x	x	x			x						x	x	x	x	x	x			
ERLING						x	x		x															
ERRIN			x			x																		
FILEM	x	x				x	x			x														
GAM						x																		
GAMM							x		x															
GASP	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			
GEN	x		x	x	x	x																		
GETIA		x					x								x									
GETPR							x			x														
GETRA							x				x				x									
GETSA							x								x									
GETTC							x			x		x												
GTCHAR	x		x	x						x		x												
HISTO						x	x							x										
IMODFN	x		x	x			x						x											
INIT	x	x	x			x	x			x	x	x		x									x	
INITS	x				x	x	x	x		x						x	x		x	x				
IRATT	x		x	x			x		x		x				x							x		
ISATT	x		x	x			x		x						x							x		
LLABL	x		x	x	x		x				x						x							
LPACK				x		x																		
MAP			x	x																				
MODFN	x		x	x			x			x			x											
MONIT	x		x	x	x	x	x												x					
MSWT	x		x	x															x					
MTASK	x		x	x															x					
NFIND		x																						
NMOD	x		x	x						x														
NPOSN						x	x		x															
OUTPT	x		x	x				x																
PERTXF									x															

Subprogram Name	COMMON Block Number																							
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
PLOTS	x		x	x			x														x			
PNABA	x		x	x								x												
POP	x		x	x			x				x													
PRIOR							x			x														
PRNTH						x	x	x		x				x										
PRNTS					x	x											x							
PROB	x		x	x		x	x			x		x												
PUTIA		x					x								x									
PUTPR							x			x														
PUTRA							x				x				x									
PUTSA							x								x									
PUTTC							x			x		x												
QRANK							x																	
RCLEAR	x		x	x			x			x														
REG	x		x	x	x					x								x						
RLOGN							x		x															
RMOVE	x	x				x																		
RNORM						x	x		x															
RPLOT	x				x	x	x	x												x	x			
SCHAT		x				x	x	x		x	x	x	x	x	x							x		
SCHED		x				x	x			x	x													
SCOND						x	x									x	x		x					
SEER	x	x	x		x	x	x			x					x									
SGEN	x		x	x	x	x																		
SSAVE	x					x	x	x									x			x	x			
SSTAT	x		x	x	x		x													x				
STATT	x		x	x			x			x				x										
SUMRY					x	x	x	x		x	x			x										
TASK	x		x	x			x		x	x	x													
TCLEAR	x		x	x						x														
TIMEQ		x				x																		
TMARK		x					x								x									
TRNGL						x	x		x															

Subprogram Name	COMMON Block Number																							
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
UCLCT						x	x																x	
UCOLL	x		x	x			x																x	
UHIST						x	x																x	
UHSTO	x		x	x			x																x	
UNFRM						x	x		x															
UPDATE		x			x	x	x	x		x		x		x	x	x	x	x	x					
UPLOT	x				x	x	x																	x
UPLTS	x		x	x			x																	x
UTIME	x		x	x			x																x	
UTMSA							x																x	
UTMST						x	x																x	
UVAR	x		x	x																				x
VAR	x		x	x	x																x			
WEIBL						x	x		x															

REFERENCES

1. Wortman, D.B., S.D. Duket, D.J. Seifert, R.L. Hann, and G.P. Chubb, Simulation Using SAINT: A User-Oriented Instruction Manual, AMRL-TR-77-61, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

2. Duket, S.D., D.B. Wortman, D.J. Seifert, R.L.Hann, and G.P. Chubb, Documentation for the SAINT Simulation Program, AMRL-TR-77-63, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

3. Duket, S.D., D.B. Wortman, D.J. Seifert, R.L. Hann, and G.P. Chubb, Analyzing SAINT Output Using SPSS, AMRL-TR-77-64, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

BIBLIOGRAPHY

Askren, W.B., W.B. Campbell, D.J. Seifert, T.J. Hall, R.C. Johnson, and R.H. Sulzen, Feasibility of a Computer Simulation Method for Evaluating Human Effects on Nuclear Systems Safety, AFWL-TR-76-15, Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, May 1976.

Chubb, G.P., "The Use of Monte Carlo Simulation to Reflect the Impact Human Factors Can Have on Systems Performance," Proceedings of the 1971 Winter Simulation Conference (Fifth Conference on the Applications of Simulation), held 8-10 December 1971, New York, sponsored by ACM/AIIE/IEEE/SHARE/SCI/TIMS, also identified by AMRL-TR-71-75.

Chubb, G.P., "Using Monte Carlo Simulation to Assess the Impact Radiation Induced Performance Degradation Can Have on Mission Success," presented at the Navy Symposium on Computer Simulation as Related to Manpower and Personnel Planning, Naval Personnel Research and Development Laboratories, Washington, D.C., July 1971.

Clayton, E.R., and L.J. Moore, "GERT vs. PERT," Journal of System Management, Vol. 23, February 1972, pp. 18-19.

Duket, S.D., D.B. Wortman, and D.J. Seifert, SAINT Simulation of a Remotely Piloted Vehicle/Drone Control Facility: Technical Documentation, AMRL-TR-75-119, AD A-029944, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

Emshoff, J.R., and R.L. Sisson, Design and Use of Computer Simulation Models, New York: The Macmillan Company, 1970.

Evans, G.W., II, G.F. Wallace, and G.L. Sutherland, Simulation Using Digital Computers, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1967.

Fishman, G.S., Concepts and Methods in Discrete Event Digital Simulation, New York: John Wiley and Sons, Inc., 1973.

Gordon, G., Systems Simulation, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1969.

Hann, R.L., and G.G. Kuperman, "SAINT Model of a Choice Reaction Time Paradigm," Proceedings of the 19th Annual Meeting of the Human Factors Society, AMRL-TR-75-25, AD A-027931, Dallas, October 1975.

Kuperman, G.G., and D.J. Seifert, "Development of a Computer Simulation Model for Evaluating DAIS Display Concepts," Proceedings of the 19th Annual Meeting of the Human Factors Society, Dallas, October 1975.

Maltas, K.L., and J.R. Buck, "Simulation of a Large Man/Machine Process Control System in the Steel Industry," Proceedings of the 19th Annual Meeting of the Human Factors Society, Dallas, October 1975.

Mirham, G.A., Simulation: Statistical Foundations and Methodology, New York: Academic Press, 1972.

Mize, J.J., and J.G. Cox, Essentials of Simulation, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1968.

Moder, J.J., and C.R. Phillips, Project Management with CPM and PERT, New York: Prinhold Publishing Corporation, 1964.

Moore, L.J., and E.R. Clayton, GERT Modeling and Simulation: Fundamentals and Applications, New York: Petrocelli/Charter, 1976.

Naylor, T.H., J.L. Balintfy, D.S. Burdick, and K. Chu, Computer Simulation Techniques, New York: John Wiley & Sons, Inc., 1965.

Pritsker, A.A.B., The GASP IV User's Manual, Pritsker & Associates, Inc., West Lafayette, Indiana, 1973.

Pritsker, A.A.B., The GERTE User's Manual, Pritsker & Associates, Inc., West Lafayette, Indiana, 1974.

Pritsker, A.A.B., The P-GERT User's Manual, Pritsker & Associates, Inc., West Lafayette, Indiana, 1974.

Pritsker, A.A.B., The Q-GERT User's Manual, Pritsker & Associates, Inc., West Lafayette, Indiana, 1974.

Pritsker, A.A.B., and P.J. Kiviat, Simulation with GASP II, A FORTRAN-Based Simulation Language, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1969.

Pritsker, A.A.B., and C.E. Sigal, The GERT IIIZ User's Manual, Pritsker & Associates, Inc., West Lafayette, Indiana, 1974.

Pritsker, A.A.B., D.B. Wortman, G.P. Chubb, and D.J. Seifert, "SAINT: Systems Analysis of Integrated Networks of Tasks," Proceedings of the Fifth Annual Pittsburgh Conference on Modeling and Simulation, Pittsburgh, Pa.: April 1974.

Pritsker, A.A.B., D.B. Wortman, C.S. Seum, G.P. Chubb, and D.J. Seifert, SAINT: Volume I. Systems Analysis of Integrated Networks of Tasks, AMRL-TR-73-126, AD A-014843, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, April 1974.

Pritsker, A.A.B., and C.E. Sigal, The GERT IIIZ User's Manual, Pritsker & Associates, Inc., West Lafayette, Indiana, 1974.

Schmidt, J.W., and R.E. Taylor, Simulation and Analysis of Industrial Systems, Homewood, Ill.: Richard D. Irwin, Inc., 1970.

Schriber, T.J., Simulation Using GPSS, New York: John Wiley & Sons, Inc., 1974.

Seifert, D.J., and G.P. Chubb, Computer Models of Man-Machine Survivability/Vulnerability, AMRL-TR-72-69, AD 762528, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, April 1973.

Shannon, R.E., Systems Simulation: The Art and Science, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1975.

Siegel, A.I., and J.J. Wolf, Man-Machine Simulation Models, New York: John Wiley & Sons, Inc., 1969.

Siegel, A.I., J.J. Wolf, and R.T. Sorenson, Techniques for Evaluating Operator Loading in Man-Machine Systems: Evaluation of a One or a Two Operator Evaluative Model Through a Controlled Laboratory Test, Applied Psychological Services, Inc., Wayne, Pa., 1962.

Siegel, A.I., J.J. Wolf, M.A. Fischl, W. Miehle, and G.P. Chubb, Modification of the Siegel-Wolf Operator Simulation Model for On-Line Experimentation, AMRL-TR-71-60, AD 737798, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, June 1971.

Sigal, C.E., and A.A.B. Pritsker, "SMOOTH: A Combined Continuous-Discrete Network Simulation Language," SIMULATION, March 1974.

Sigal, C.E., "SMOOTH: A Combined Continuous-Discrete Network Simulation Language," Unpublished M.S.I.E. Thesis, Purdue University, West Lafayette, Indiana, August 1973.

Townsend, T., "GERT Networks with Item Differentiation Capabilities," Unpublished M.S.I.E. Thesis, Purdue University, West Lafayette, Indiana, August 1973.

Whitehouse, G.E., Systems Analysis and Design Using Network Techniques, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1973.

Wortman, D.B., S.D. Duket, and D.J. Seifert, "Simulation of a Remotely Piloted Vehicle/Drone Control Facility Using SAINT," Proceedings of the 1975 Summer Computer Simulation Conference, San Francisco, July 1975.

Wortman, D.B., S.D. Duket, and D.J. Seifert, New Developments in SAINT: The SAINT III Simulation Program, AMRL-TR-75-117, AD A-029894, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, 1975.

Wortman, D.B., S.D. Duket, and D.J. Seifert, "SAINT Simulation of a Remotely Piloted Vehicle/Drone Control Facility," Proceedings of the 19th Annual Meeting of the Human Factors Society, Dallas, October 1975.

Wortman, D.B., A.A.B. Pritsker, C. Seum, D.J. Seifert, and G.P. Chubb, SAINT: Volume II. User's Manual, AMRL-TR-128, AD A-011586, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, April 1974.

Wortman, D.B., C.E. Sigal, A.A.B. Pritsker, and D.J. Seifert, New SAINT Concepts and the SAINT II Simulation Program, AMRL-TR-74-119, AD A-014814, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, April 1975.

Wortman, D.B., C.E. Sigal, A.A.B. Pritsker, and D.J. Seifert, SAINT II Documentation Manual, AMRL-TR-75-116, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, April 1975.

Wortman, D.B., S.D. Duket, and D.J. Seifert, SAINT Simulation of a Remotely Piloted Vehicle/Drone Control Facility: Model Development and Analysis, AMRL-TR-75-118, AD A-031085, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

*U.S.Government Printing Office: 1978 — 757-080/781