

AD-A121 987

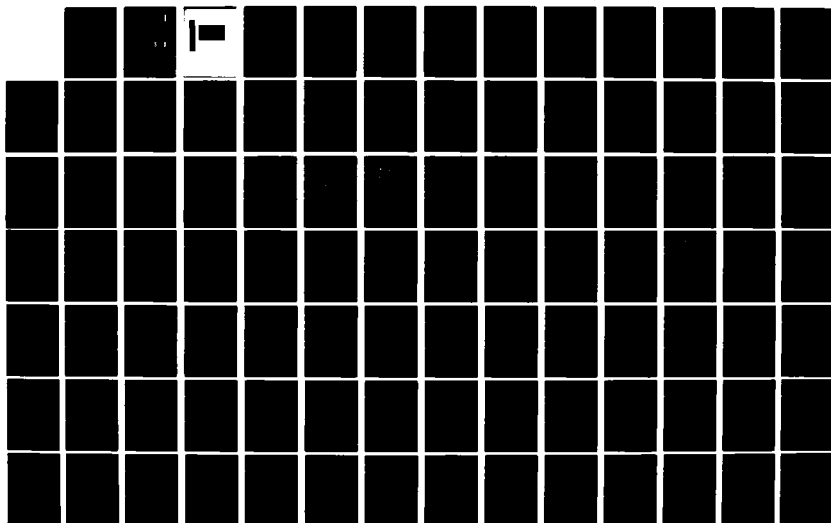
RIME: THE RECOVERABLE ITEM MANAGEMENT EVALUATOR VOLUME
II SECTION III RIM. (U) DECISION SYSTEMS DAYTON OH
W S DENNY MAY 80 TR-80-02-C F33600-78-C-0524

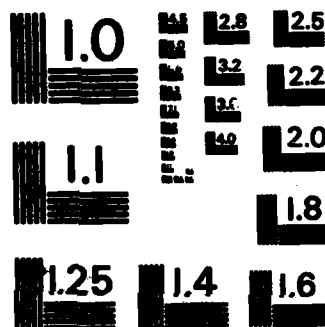
1/2

UNCLASSIFIED

F/G 9/2

NL



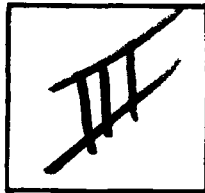


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

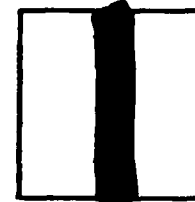
PHOTOGRAPH THIS SHEET

AD A 121987

DTIC ACCESSION NUMBER



LEVEL



INVENTORY

RIME: Vol. II, Section III

DOCUMENT IDENTIFICATION

Rept. No. TR-80-02-C

Contract F33600-78-C-0524

May 80

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR

NTIS GRA&I

DTIC TAB

UNANNOUNCED

JUSTIFICATION



BY

DISTRIBUTION /

AVAILABILITY CODES

DIST

AVAIL AND/OR SPECIAL

A

DISTRIBUTION STAMP

DTIC
ELECTE
DEC 1 1982
D

DATE ACCESSIONED



DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

RIME:

The Recoverable Item Management Evaluator:

Volume II, Section III

RIME Simulation Model Programs

RIME:
The Recoverable Item Management Evaluator:
Volume II, Section III
RIME Simulation Model Programs

by
W. Steven Demmy

May 1980

TR-80-02-C

DECISION SYSTEMS
2125 Crystal Marie Drive
Dayton, Ohio 45431

(513) 426-8515

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

82 11 30 027

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RIME: The Recoverable Item Management Evaluator: Volume II: Program Listings and Narratives		5. TYPE OF REPORT & PERIOD COVERED INTERIM August 79-Jun 80
7. AUTHOR(s) W. Steven Denny		6. PERFORMING ORG. REPORT NUMBER TR-80-02-C
9. PERFORMING ORGANIZATION NAME AND ADDRESS Decision Systems 3575 Charlene Drive Dayton, Ohio 45432		8. CONTRACT OR GRANT NUMBER(s) C-0524 F33600-78-D-0214
11. CONTROLLING OFFICE NAME AND ADDRESS 2750th ABW/PMA BLDG 1, Area C Wright-Patterson AFB, Ohio		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 23041A5
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE May 1980
		13. NUMBER OF PAGES 298
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
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) A		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Recoverable item, multi-echelon, inventory/repair, simulation, METRIC, MOD-METRIC, AFLC		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the Recoverable Item Management Evaluator (RIME), a FORTRAN simulation model for evaluating the relative cost-effectiveness of analytic optimization procedures proposed for use in Air Force Logistics Command recoverable item management systems. Major features of the model include (a) the use of actual Air Force demand histories to drive the model demand processes, (b) modeling of current Air Force statistical estimation procedures, and (c) modeling of the dynamic interactions among initial provisioning, replenishment and distribution policies. Volume II documents the programs for RIME.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Section III

RIME Simulation Model Programs

List of Programs

Program

BWAIT

CONDEM

CUM, CUMB

ENTER

ENTERB

EVNTS

FILLBO

FILLST

INGASP

INFEL

INITAL

INITEM, INITM1, INITM2

ITRSLT

KNSKU

LEVEL

LEVEL2

Program

NRTS

ORDER

OUT2

OUTREP

RCVPRT

RECEIV

REQ

REMOVE

RIME

RIMEB

REVIEW

REPGEN

REQ

SSTAT

STATHD

WRIFEL

ZERO

Subroutine: BWAIT**Functions**

Event code 16. This routine simulates an LRU entering a Wait-for-parts status.

Calling Parameters:

N = Stock Keeping Unit Number of the LRU

NNEED = The total number of SRU components required for this LRU

NJOB = Job Number for this particular LRU reparable generation

Description

This routine simulates an LRU entering a wait-for-parts status. To do this, it loads the values of N, NNEED, NJOB, and ITIME into the vector ATRIB, where ITIME is the current clock time. The GASP routine FILEM is then called to record these data values in GASP file 2.

Subroutine RCVPRT (event code 17) eventually removes this LRU from file 2 after all NNEED SRUs needed for the repair of the LRU have been received.

04 10-20-79 10,304

SUBROUTINE BWAIT(N,NNEED,NJOB)

EVENT CODE 46. THIS ROUTINE SIMULATES AN LRU ENTRY
WAIT-FOR-PARTS STATUS.

```

COMMON /GCOM1/ ATRIB(30),JEVNT,MFA,MFE(100),MLE(100),MSTOP,MCRDR,
NWAP0,NWAPT,NNATR,NWPL,NWQ(100),NNWRY,NPNT,PPARM(50,4),TNOW,TTBEG,
TTCLR,TTFIN,TTIR(20),TTSET
COMMON /GCOM6/ BENG(100),IINN(100),KKRNN(100),MMAXO(100),
QOTIM(100),ESOBV(25,5),ESTPV(25,6),VYNC(100)
DIMENSION NSET(1)
COMMON OSET(1)
EQUIVALENCE (NSET(1),OSET(1))
COMMON /ITIME /ITIME

```

FILE NJOB IN THE WAIT-FOR-PARTS FILE

```

ATRIB(1) = N
ATRIB(2) = NNEED
ATRIB(3) = NJOB
ATRIB(8) = ITIME
CALL FILEM (2)
RETURN
END

```

BWAIT

Subroutine: CONDEM

Function:

Event code 15. This routine records the condemnation of IQTY units of SKU N. NJOB is the associated reparable generation number.

Calling Parameters:

N = Stock Keeping Unit number
IQTY = The number of units to be condemned
NJOB = The reparable generation number associated with this condemnation

Description:

This routine calls subroutine CUM to update condemnation statistics. It then updates the number of assets which are work-in-process (INWIP), and returns to the calling routine.

```
1      SUBROUTINE CONDEM(N,IQTY,NJOB)
2      C      EVENT CODE 15.  THIS ROUTINE RECORDS THE CONDEMNATION
3      C      ON IQTY UNITS OF SKU N; NJOB IS THE ASSOCIATED
4      C      REP GEN NUMBER.
5      C
6      COMMON/ICNDEM/ICNDEM(16,3,6)
7      COMMON/INWIP/INWIP(1)
8      C
9      C      UPDATE CONDEMNATION STATUS
10     C
11     C      CALL CUM(ICNDEM,IQTY,N)
12     C
13     C      UPDATE WORK-IN-PROCESS
14     C
15     C      INWIP(N)=INWIP(N)-IQTY
16     C
17     C
18     C      RETURN
19     C      END
```

CONDEM

Subroutine: CUM (ISTAT, IQTY, KFSN)

Function:

This routine updates the items, units, and dollar statistics associated with the statistics array ISTAT.

Calling Parameters:

- ISTAT = The statistics array to be updated
- IQTY = The number of units associated with this transaction
- KFSN = The statistics collection index to be updated. See Volume I for a definition of allowable values for KFSN.

Subroutine: CUMB (ISTAT, IQTY, KFSN)

Function:

This routine updates the unit and dollar counts associated with the statistics array in ISTAT. This routine has no effect upon the counts of FSN actions.

Calling Parameters:

- ISTAT = Statistic array to be updated
- IQTY = The number of units associated with this transaction
- KFSN = The statistics collection index

01 10-20-79 10.529 CUMULATE SIMULATION STATISTICS

```

1  *SRUN=1/INHS/688/CUM.O(BCD,NO60)
2  *CUM.S      CUMULATE SIMULATION STATISTICS
3  SUBROUTINE CUM(ISTAT,IQTY,KPSN)
4      C      THIS ROUTINE UPDATES THE ITEMS, UNITS, AND DOLLARS
5      C      ASSOCIATED WITH ONE STATISTIC ISTAT
6      C
7      DIMENSION ISTAT(16,3,6)
8      COMMON/XTINV/XTINV
9      COMMON/UCOST/UCOST(1)
10     COMMON/IGBUS/IGBUS
11     C
12     I=XTINV
13     C
14     C      COMPUTE K (STATISTIC COLLECTION INDEX), WHERE
15     C      K=1 DENOTES LRU AT BASE
16     C      K=2 DENOTES SRU AT BASE
17     C      K=3 DENOTES LRU AT DEPOT
18     C      K=4 DENOTES SRU AT DEPOT
19     C      K=5 DENOTES LRU AT OVERHAUL
20     C      K=6 DENOTES SRU AT OVERHAUL
21     C
22     K=KNSKU(KPSN)
23     C
24     C      UPDATE COUNT OF PSN ACTIONS
25     C
26     IF (IQTY.NT.0) GO TO 21
27     ISTAT(I,1,K)=ISTAT(I,1,K)+1
28     GO TO 28
29     21 CONTINUE
30     ISTAT(I,1,K)=ISTAT(I,1,K)+1
31     GO TO 22
32     C
33     C      UPDATE UNIT AND DOLLAR COUNTS. NOTE THAT ENTRY
34     C      POINT CUMB HAS NO EFFECT ON PSN ACTION COUNTS.
35     C
36     ENTRY CUMB(ISTAT,IQTY,KPSN)
37     I=XTINV
38     K=KNSKU(KPSN)
39     22 ISTAT(I,2,K)=ISTAT(I,2,K)+IQTY
40     DOLLAR=UCOST(KPSN)*FLOAT(IQTY)
41     ISTAT(I,3,K)=ISTAT(I,3,K)+DOLLAR+.5)
42     C
43     IF (IGBUS.NE.1) RETURN
44     WRITE(6,23) KPSN,IQTY,I,K,(ESGAB(I,J,K),J=1,3)
45     23 FORMAT(8('***'), 'CUM=KPSN=P.15,' XTINV,
46     & '15.' XTINV=P.15, ' K=',I5,' UPDATED STATS=P.318)
47     C
48     RETURN
49     END

```

CUM.S

Subroutine: ENTER

Function:

This routine enters transactions on the Future Events List, and updates the associated pointer variables.

Description:

Subroutine ENTER places events on the Future Events List, and updates the associated pointer variables. These transactions are subsequently removed by subroutine REMOVE. See Volume I, Section II for a detailed description of the operations of these routines.

01 10-20-79 10.421 .S

```

1  *NRUN=TIME/OBJ/FELIST.O(BGD.MQGO)
2  *FELIST,S
3  SUBROUTINE ENTER(KTIME,KTYPE,KFSN,KQTY,KPRIOR)
4  C      THIS ROUTINE ENTERS EVENTS ON THE FUTURE EVENTS LIST
5  C      AND UPDATES THE CHAIN STRUCTURE
6  COMMON/IDBUG/IDBUG
7  COMMON/ITIME/ITIME
8  COMMON/NENTRY/NENTRY
9  COMMON/NEMAX/NEMAX
0  COMMON/NFIRST/NFIRST
1  COMMON/NLOC/NLOC
2  COMMON/NTIME/NTIME
3  COMMON/ILOCFF/ILOCFF(1)
4  COMMON/JFSN/JFSN(1)
5  COMMON/JPOINT/JPOINT(1)
6  COMMON/JPRIOR/JPRIOR(1)
7  COMMON/JQTY/JQTY(1)
8  COMMON/JTIME/JTIME(1)
9  COMMON/JTYPE/JTYPE(1)
0  C
1  C      CHECK IF EVENT TIME EXCEEDS 16 OTS(134400 TIME UNITS).
2  C      IF SO, GO TO STATEMENT 100 AND RETURN.
3  C
4  IF(KTIME.GT.134400) GO TO 100
5  C
06 10 CONTINUE
07 C      UPDATE NO. OF ENTRIES ON THE F.E.L
08 C
09 NENTRY=NENTRY+1
10 C      IF THIS ENTRY EXCEEDS CAPACITY OF THE CHAIN,PRINT
11 C      ERROR MESSAGE AND QUIT
12 IF(NENTRY.GT.NEMAX) GO TO 90
13 C      PLACE TRANSACTION IN FIRST AVAILABLE LOCATION
14 20 K=ILOCFF(NENTRY)
15 NLOC=K
16 JTIME(K)=KTIME
17 JTYPE(K)=KTYPE
18 JFSN(K)=KFSN
19 JQTY(K)=KQTY
20 JPRIOR(K)=KPRIOR
41 C      CHECK TO SEE IF THIS IS THE ONLY ENTRY ON THE LIST
42 IF(NENTRY.NE.1) GO TO 40
43 C      UPDATE LIST STATUS AND POINTER VARIABLES
44 NFIRST=K
45 NTIME=KTIME
46 JPOINT(K)=0
47 GO TO 100
48 C      DOES THE NEW TRANSACTION PRECEED ALL OTHER LIST ENTRIES(
49 40 IF(KTIME.GE.NTIME) GO TO 60
50 C      INSERT NEW TRANSACTION AS A FIRST LINK IN THE CHAIN
51 NTIME=KTIME
52 JPOINT(K)=NFIRST

```

ENTER

P 01 10-20-79 10,421 .S

```

53      NFIRST=K
54      GO TO 100
55      C      THE FOLLOWING STEPS SEARCH DOWN THE FEL AND
56      C      INSERTS THE TRANSACTION IN THE PROPER LOCATION
57      60  JG=NFIRST
58      KS=JPOINT(JJ)
59      C      IS JJ THE LAST RECORD ON THE CHAIN?
60      62  IF(KJ,80,0) GO TO 97
61      C      DOES THE NEW TRANSACTION PRECEED KJ?
62      IF(KTIME,LT,JTIME(KS)) GO TO 80
63      JS=KJ
64      KS=JPOINT(JJ)
65      GO TO 62
66      C      INSERT      NEW TRANSACTION      AS THE LAST LINK ON THE CHAIN
67      97  JPOINT(JJ)=K
68      JPOINT(K)=0
69      GO TO 100
70      C      INSERT      NEW TRANSACTION      ON THE CHAIN
71      80  JPOINT(JJ)=K
72      JPOINT(K)=KJ
73      GO TO 100
74      C      WRITE ERROR MESSAGE
75      90  WRITE(6,91)NPEMAX
76      91  FORMAT(//,"CAPACITY OF FUTURE EVENTS LIST EXCEEDED",
77      &      //" CURRENT CAPACITY=",I5//)
78      CALL WRIFEL
79      STOP
80      100 CONTINUE
81      IF(IDBUG,NE,1) GO TO 25
82      WRITE(6,8000)ITIME,KTIME,KTYPE,KPSN,KQTY,KPBOR,
83      &      K,NFIRST,NENTRY,NTIME
84      8000 FORMAT(7H ENTER ,I10.40X,8I8,I7)
85      25  CONTINUE
86      RETURN
87      END

```

Subroutine: ENTERB**Function:**

This routine records backorders in the backorder file, and updates associated pointer variables.

Calling Parameters

- N = The Stock Keeping Unit number associated with current backorder.
- IQTY = The number of units to be backordered.
- LPRI = A packed variable which defines both the priority of the requisition, and the SKU identification of the stocking location which originated the requisition.
- JTIME = The time (in TMUs) that the requisition is entered into the backorder file.

Description:

This routine records backorders in the backorder file. Backorders are stored in linked list form, with pointers from each recorded backorder to the next lower priority outstanding backorder for the associated item. In this linked list, the highest priority, oldest backorder is stored first; that is, backorders are recorded on a first-in-first-out basis within priority. See Volume I, Section II for variable definitions.

The variable NBOTP (N) defines the location of the oldest, highest priority backorder for SKU N. When new stock is received, this backorder will be the first requisition to be filled.

As noted above, the calling parameter LPRI defines both the priority of the backordered requisition, and the Stock Keeping Unit number of the inventory location which originated the requisition. The last two digits of LPRI define the priority of the requisition. That is, the requisition priority IPRI is given by $\text{MOD}(\text{LPRI}, 100)$. If LPRI is less than 100, we assume the requisition was submitted by a flight-line maintenance group located at Stock Keeping Unit N. If LPRI is greater than 100, but less than 1000, we assume the backorder originated from a replenishment request from some stocking location other than location N. In this case, the Stock Keeping Unit number of the requisitioning organization is given by $(\text{LPRI} - \text{IPRI})/100$. If the parameter LPRI is greater than 1000, the backorder represents a requisition for SRU components needed to complete the repair of an LRU. In this case, the job number, NJOB, associated with the LRU repair is given by $\text{NJOB} = (\text{LPRI} - \text{IPRI})/100$.

When Subroutine ENTERB is called, the routine first decodes LPRI to determine the stock keeping unit and priority associated with this requisition. It next updates the backorder counters NBOTR (N), NBOTU (N), NBOIR (N), and NBOIU (N) which define the total current requisition and unit backorders for SKU N. Finally, the routine finds the appropriate location in the linked list to record this new backorder, and updates the associated pointer variables.

```

1  SUBROUTINE ENTERB(N,IQTY,IPRI,JTIME)
2  SUBROUTINE ENTERB(N,IQTY,IPRI,JTIME)
3  C THIS ROUTINE BACKORDERS REQUISITIONS FOR ITEM N
4  C IQTY=QUANTITY PLACED ON BACKORDER
5  C IPRI=1 HIGH-PRIORITY REQUISITION
6  C IPRI=2 OTHERWISE
7  C IF IPRI > 1000, THEN LPRI=100*NIJOB + IPRI
8  C
9  C JTIME= CLOCK TIME REQ WAS RECEIVED
0  COMMON/IDBUG/IDBUG
1  COMMON/NBMAX/NBMAX
2  COMMON/NLOCBK/NLOCBK
3  COMMON/NBOIU/NBOIU(1)
4  COMMON/NBOIR/NBOIR(1)
5  COMMON/NBOTR/NBOTR(1)
6  COMMON/NBOTU/NBOTU(1)
7  COMMON/NBOPT/NBOPT(4)
8  COMMON/IBACPT/IBACPT(1)
9  COMMON/IDFSNB/IDFSNB(1)
0  COMMON/ILOCBK/ILOCBK(1)
1  COMMON/IPRIOR/IPRIOR(1)
2  COMMON/IQTYE/IQTYE(1)
3  COMMON/ITMBAC/ITMBAC(1)
4  C COMPUTE PRIORITIES AND STOCK KEEPING UNIT
5  C
6  NSKU=N
7  IPRK=MOD(IPRI,100)
8  IF(LPRI.GT.100)NSKU=(LPRI-IPRI)/100
9  C RESERVE A STORAGE LOCATION FOR THIS INFORMATION
0  IPT=ILOCBK(NLOCBK)
1  C
2  C UPDATE BACKORDER COUNTERS
3  C
4  NBOTR(N)=NBOTR(N)+1
5  NBOTU(N)=NBOTU(N)+IQTY
6  IF(IPRI.NE.1)GO TO 10
7  NBOIR(N)=NBOIR(N)+1
8  NBOIU(N)=NBOIU(N)+IQTY
9  10 CONTINUE
0  IF(IDBUG.NE.1)GO TO 15
1  WRITE(6,13)N,NSKU,IQTY,IPRI,
2  NBOIU(N),NBOTU(N),NBOIR(N),NBOTR(N),IPT
3  13 FORMAT(4X,'***ENTERB--N=',I5,' NSKU=',I5,' IQTY=',I5,
4  ' IPRI=',I5,
5  ' NBOIU=',I5,' NBOTU=',I5,' NBOIR=',I5,' NBOTR=',I5,
6  ' IPT=',I5)
7  15 CONTINUE
8  C DID THIS REQUISITION CAUSE THE BACKORDER FILE TO OVERFLOW
9  NLOCBK=NLOCBK+1
0  IF(NLOCBK.GE.0)GO TO 20
1  C WRITE ERROR MESSAGE
2  WRITE(6,91)

```

ENTERB

```

3      91 FORMAT(/20X,"ERROR=BACKORDER FILE OVERFLOW. ")
4      WRITE(6,82)
5      82 FORMAT(1H1,10X,23H"*BACKORDER FILE DUMP*")
6      DO 83 K=1,NBMAX
7      83 WRITE(6,84) K,ITMBAC(K),IDFSNB(K),IPRIOR(K),IQTYB(K),IBACPT(K)
8      84 FORMAT(/3X,"REC NO=",I3,5X,"ITMBAC=",I7,5X,"IDFSNB=",I10,5X,
9      & 7HPRIOR=,I1,5X,6HIQTYB=,I7,5X,7HIBACPT=,I9)
10     STOP
11     C      RECORD QUANTITY,PRIORITY,FSN ID,AND TIME DATA FOR THIS BO REQ
12     20 ITMBAC(IPT)=JTIME
13     IDFSNB(IPT)=NSKU
14     IPRIOR(IPT)=IPRI
15     IQTYB(IPT)=IQTY
16     C      ARE ANY OTHER BACKORDERS OUTSTANDING ON ITEM N
17     IF(NBOPT(N).GT.0) GO TO 40
18     C      RECORD POINTER DATA
19     NBOPT(N)=IPT
20     IBACPT(IPT)=0
21     RETURN
22     C      IS THE NEW BO A PRIORITY 1 REQUISITION
23     40 IF(IPRI.EQ.1) GO TO 60
24     C      NOTE-- NEW LOW PRIORITY BACKORDERS ARE INSERTED LAST ON THE
25     C      BACKORDER CHAIN, THE REMAINING STEPS IN THIS PORTION
26     C      OF THE SUBROUTINE ACCOMPLISH THIS OBJECTIVE
27     C      SET JPT EQUAL TO THE FILE LOCATION NO OF THE FIRST BACKORDERED
28     C      REQUISITION IN THE CHAIN
29     JPT=NBOPT(N)
30     C      IS JPT THE LAST LINK IN THE CHAIN
31     49 IF(IBACPT(JPT).EQ.0) GO TO 50
32     KPT=IBACPT(JPT)
33     JPT=KPT
34     GO TO 49
35     C      RECORD NEW POINTERS
36     50 IBACPT(JPT)=IPT
37     IBACPT(IPT)=0
38     RETURN
39     C      SET JPT EQUAL TO LOCATION NO OF FIRST BO ON CHAIN
40     60 JPT=NBOPT(N)
41     C      IS BACKORDER JPT A HIGH PRI BO
42     61 IF(IPRIOR(JPT).NE.1) GO TO 80
43     C      IS JPT THE LAST BACKORDER ON THE CHAIN
44     IF(IBACPT(JPT).EQ.0) GO TO 62
45     KPT=JPT
46     JPT=IBACPT(KPT)
47     GO TO 61
48     C      INSERT NEW BO AS LAST LINK ON CHAIN
49     62 IBACPT(JPT)=IPT
50     IBACPT(IPT)=0
51     RETURN
52     C      IS JPT THE ONLY BO ON CHAIN
53     80 IF(JPT.NE.NBOPT(N)) GO TO 81
54     C      INSERT NEW BO AS FIRST LINK ON CHAIN

```

```
15      NROPT(N)=IPT
16      IBACPT(IPT)=JPT
17      RETURN
18      C      INSERT      NEW SO AS LINK BETWEEN KPT AND JPT
19      S1 IBACPT(KPT)=IPT
20      IBACPT(IPT)=JPT
21      RETURN
22      END
```

Subroutine: EVNTS**Function:**

This routine takes events off of the Future Events List and the Exogenous Event File based on the lowest event time, but then calls the appropriate event subroutine.

Calling Parameters:

RNLAST = This parameter specifies the random number seed to be used for initialization of the pseudo-random number generator.

Description:

Subroutine EVNTS controls the timing of individual events throughout a simulation run. The routine begins by initializing the pseudo-random number generator, the Future Events List and associated timing variables, and by initializing the data arrays for the LRU/SRU family group to be simulated. This is accomplished through calls to subroutines RANDU, INITAL, and INITM2, respectively. In program lines 480 through 810, the routine then determines whether the next most eminent event, (i.e., the event with the lowest scheduled clock time) is on the Exogenous Event File or on the Future Events List. It then removes this event from the appropriate file, and sets the parameter values for the current event. In program line 840, the routine then branches to the appropriate section of program logic associated with the event type of the current event. If an End-of-Run event is encountered (event type 10), program control returns to the MAIN program. Otherwise, the program logic returns to program line 480. The process of determining the next most eminent event and calling appropriate event subroutine logic is then repeated until a type 10 event is eventually encountered.

When a type 10 event is finally encountered, the program logic first checks (program line 1470) if this event was read from the Exogenous Event File. If so, the program immediately returns to the MAIN program. Otherwise, Subroutine EVNTS continues to read events from the Exogenous Event File until a type 10 event is eventually encountered. When this happens, program control returns to the MAIN program. This last, "tidy up" process is required to ensure that when the simulation of the next LRU/SRU group begins, the Exogenous Event File is positioned at the beginning of the events list that applies to the new group.

01 10-20-79 10.536 E

```

1  *RUN=JTIME/053/EVNTS.0(NCD,NOSO)
2  *EVNTS.S
3  SUBROUTINE EVNTS(NLAST)
4  C      THIS ROUTINE TAKES EVENTS OFF THE FUTURE EVENTS LIST
5  C      AND THE EXOGENOUS EVENT FILE BASED ON LOWEST EVENT
6  C      TIME. IT THEN CALLS ONE APPROPRIATE EVENT SUBROUTINE.
7  C
8  COMMON/ETIME/ETIME
9  COMMON/ETIME/ETIME
10 COMMON/IDBUG/IDBUG
11 COMMON/ITRACE/ITRACE,ISTRAC
12 COMMON/NITEM/NITEM
13 COMMON/INACT/INACT(1)
14 COMMON/INVDUE/INVDUE(1)
15 COMMON/INWIP/INWIP(1)
16 COMMON/NBOTR/NBOTR(1)
17 COMMON/NBOTR/NBOTR(1)
18 COMMON/NBOTU/NBOTU(1)
19 COMMON/NBOTU/NBOTU(1)
20 C
21 COMMON/NREPL/NREPL
22 C
23 C*****
24 C
25 C      INITIALIZE RANDOM NUMBER GENERATOR
26 C
27 X=-NRLNST
28 CALL RANDU(X)
29 15 CONTINUE
3  C      INITIALIZE THE FUTURE EVENTS LIST(INITIAL).
31 C
32 CALL INIPL
33 C
34 C      INITIALIZE DATA ARRAYS FOR THIS ITEM
35 C
36 CALL INIT2
37 C
38 C      READ FIRST ENTRY ON EXOG. EVENT FILE
39 CC
40 READ(7)JTIME,JTYPE,JF3,JF4,JF5
41 C
42 IF(IDBUG .NE. 1) GO TO 10
43 CALL WRITEL
44 16 CONTINUE
45 C
46 C*****
47 C
48 100 CONTINUE
49 C
50 C      IF NEXT EVENT IS EXOG. GO TO 150
51 C
52 IF(JTIME.LT.NTIME) GO TO 150

```

EVNTS

7T 01 10-20-79 10.938 8

```

53      C
54      C      ENDOGENOUS EVENT.  REMOVE IT FROM THE PHL.
55      C
56      CALL REMOVE(ITIME,ITYPE,IP3,IP4,IP5)
57      GO TO 180
58      C
59      C      EXOGENOUS EVENT.  SET CURRENT EVENT PARAMETERS,
60      C      THEN READ NEXT EXOGENOUS EVENT FROM FILE 7.
61      C
62      150 CONTINUE
63      ITIME=JTIME
64      ITYPE=JTYPE
65      IP3=JP3
66      IP4=JP4
67      IP5=JP5
68      READ(7,END=155)JTIME,JTYPE,JP3,JP4,JP5
69      GO TO 180
70      C
71      C      SET JTIME TO INDICATE THERE ARE NO MORE EXOGENOUS EVENTS.
72      155 CONTINUE
73      JTIME=999999999
74      160 CONTINUE
75      IF(XDNUM,80,1)WRITE(6,151)
76      151 FORMAT(1X,ITIME,ITYPE,T20," ",T26,"QTY",T35,"IP3",
77      &      T43," ",T49,"INACT EXOGENOUS EVENT",T58," ",T64," ",
78      &      T70," ",T76," ",T82," ",T88," ")
79      IF(XDNUM,80,1)
80      &      WRITE(6,153)ITIME,ITYPE,JP3,JP4,JP5,JTIME,JTIME
81      153 FORMAT(1X,T7,3X8,T98,2X7)
82      C
83      C
84      GO TO b 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
85      &      35, 36, 37, 38, 39, 40),XTYPE
86      C
87      C*****
88      C
89      21 CONTINUE
90      C      REFLECT RECEIPT OF TRANSACTION
91      C      REQUESTTOP
92      C
93      CALL REQ(IP3,IP4,IP5,XTIME)
94      CALL REVIEW(I03,07)
95      GO TO 90
96      22 CONTINUE
97      C      RECEIPT OF SHIPMENT
98      C      REFLECT REQUEST
99      CALL RECIV(IP3,IP4)
100     GO TO 90
101     23 CONTINUE
102     CALL CANCEL(IP3,IP4,IP5)
103     GO TO 90
104     C

```

: 01 10-20-79 18.534 S

```

15      C      SERVICEABLE RETURN EVENT
16      C
17      24      CALL RNT(IP3,IP4,ITIME)
18      GO TO 90
19      25 CONTINUE
20      C      STATUS      REVIEW
21      CALL REVIEW(0:0)
22      IF(IP5,ENL0) GO TO 90
23      MTIME=ITIME+IP5
24      CALL ENTER(MTIME,5,0,0,IP5)
25      GO TO 90
26      26 CONTINUE
27      C      LEVELS      COMPUTATION
28      CALL LEVEL(0)
29      CALL REVIEW(0:0)
30      IF(IP5,ENL0) GO TO 90
31      MTIME=ITIME+IP5
32      CALL ENTER(MTIME,6,0,0,IP5)
33      GO TO 90
34      27 CONTINUE
35      C      BUY GUIDELINE
36      CALL GUIDE
37      GO TO 90
38      28 CONTINUE
39      C      BUDGET      REVIEW
40      CALL BUDGET
41      GO TO 90
42      29 CONTINUE
43      C      FORECAST UPDATE
44      CALL FORUPD
45      GO TO 90
46      30 CONTINUE
47      C*****
48      C
49      C      END OF      RUN
50      C
51      C*****
52      C
53      C      READ EXOGENOUS EVENT FILE TELL A TYPE TO EVENT
54      C      IS ENCOUNTERED. MARKING THE END OF THE CURRENT
55      C      LRU/SRU GROUP DATA SET.
56      C
57      IF(JTYPE,NO.10)GO TO 99
58      READ(7)JTYPE,JTYPE,JP3,JP4,JP5
59      IF(IDRUG,NO.1)WRITE(8,0310)JTIME,JTYPE,JP3,JP4,JP5
60      0310 FORMAT('NEXT EXOS. EVENT=',E10)
61      GO TO 30
62      C
63      31 CONTINUE
64      C      SPECIAL STATISTICS
65      CALL SNTST(IP3)
66      GO TO 90

```

```
157      32 CONTINUE
158      C      DEMAND      PARAMETER UPDATE
159      CALL DEMBAR(IP3,IP4,IP5)
160      GO TO 90
161      33 CONTINUE
162      C      EVENT 13.  UPDATE TRACE PARAMETERS
163      IF(ETIME.EQ.ITRACE)IDBUG=1
164      IF(ETIME.EQ.ISTRAC)IDBUG=0
165      GO TO 90
166      C
167      C
168      C
169      C      CODE 14.  REPAIRABLE GENERATION
170      C
171      34 CONTINUE
172      CALL REGEN(IP3,IP4,IP5)
173      GO TO 90
174      C
175      C      CODE 15.  CONDEMNATION
176      C
177      35 CONTINUE
178      CALL CONDEM(IP3,IP4,IP5)
179      CALL REVIEW(IP3,07)
180      GO TO 90
181      C
182      C      CODE 16.  BEGIN WRITING FOR PARTS
183      C
184      36 CONTINUE
185      CALL HWRIT(IP3,IP4,IP5)
186      GO TO 90
187      C
188      C
189      C      CODE 17.  RECEIVE PARTS
190      C
191      37 CONTINUE
192      CALL RCVPRT(IP3,IP4,IP5)
193      GO TO 90
194      C
195      C      CODE 18.  REPAIR COMPLETION
196      C
197      38 CONTINUE
198      CALL CRSPR(IP3,IP4,IP5)
199      GO TO 90
200      C
201      C      CODE 19.  NRTS EVENT
202      C
203      39 CONTINUE
204      CALL NRTS(IP3,IP4,IP5)
205      CALL REVIEW(IP3,07)
206      GO TO 90
207      C
208      C
```

I 01 10-20-79 10.534 8

```

09      C
10      C      EVENT 20.  INITIAL PROVISIONING.
11      40 CONTINUE
12      C      INITIAL PROVISIONING EVENT:  ORDER FOR IMMEDIATE DELIVERY
13      C      SUFFICIENT STOCK TO BRING EACH LOCATION UP TO ITS
14      C      STOCK LEVEL.
15      C
16      CALL LEVEL(0)
17      CALL REVIEW(0,1)
18      90 CONTINUE
19      C
20      C
21      C*****PRINT BG STATUS
22      IF(IDBUS.EE.1) GO TO 180
23      IF((ITYPE.GE.5).AND.(ITYPE.LE.18))GO TO 180
24      N=IP3
25      WRITE(6,2993)N,INVACT(N),INVDUE(N),INWEP(N),NBOU(N),NBOUENI,
26      &      NBOYR(N),NBOYR(N)
27      2993 FORMAT(129.2F8.6I7)
28      C
29      180 CONTINUE
30      C
31      IF(IDBUS.EE.1)PRINT 93,ITIME,ITYPE,IP3,IP4,IP5,
32      &      INVACT(N),INVDUE(N),INWEP(N),NBOU(N),N+1,NITEN)
33      93 FORMAT(17.3F3.17,(124.6(3F3.2E*)) )
34      C
35      GO TO 180
36      C
37      C
38      99 CONTINUE
39      RETURN
40      END

```

Subroutine: FILLBOFunction:

This routine initiates shipping actions to fill outstanding backorders.

Calling Parameters:

N = The Stock Keeping Unit number for which outstanding backorders are to be filled.

Description:

Subroutine FILLBO (N) is called to initiate shipping actions to fill outstanding backorders for SKU N. The routine assumes that backordered requisitions are filled using a first-in, first-out, by priority, issue rule. If on hand assets are insufficient to completely fill a given requisition, partial shipments are initiated. Reduced shipments to partially fill several outstanding backorders are not permitted in this routine.

When FILLBO is called, it checks if the highest-priority outstanding requisition for SKU N may be filled completely from on-hand stock. If so, and if this shipment will not take on hand stock below the support level (ISUL(N)), the requisition is removed from the backorder list, and a shipment for the requisition quantity is initiated. If the backorder is low priority (that is, if the priority code is 2), shipments will be made until on hand stocks just equal the support levels ISUL(N). For high priority requisitions, shipments are made until on hand stock is reduced to zero or until all high priority requisitions are filled, whichever occurs first.

Subroutine FILLST is called by FILLBO to update backorder statistics and to initiate any subsequent endogenous events to be created as a result of the fill action.

01 10-20-79 14.876 14

```

1  *#RUN=ARINS/002/FILLBO.O(ACB.W000Y
2  *FILLBO.S
3      SUBROUTINE FILLBO(N)
4      COMMON/IDB00/IDB00
5      COMMON/INBOPT/INBOPT(1)
6      COMMON/INVACT/INVACT(1)
7      COMMON/IOTYS/IOTYS(1)
8      COMMON/ITNBAG/ITNBAG(1)
9      COMMON/ISUB/ISUB(1)
10     COMMON/EDESUB/EDESUB(1)
11     COMMON/ISRTOR/ISRTOR(1)
12     COMMON/ISACPT/ISACPT(1)
13     COMMON/INOCBK/INOCBK
14     COMMON/ILOCBK/ILOCBK(1)
15     25 IF(NBOPT(N).LE.0) RETURN
16     IF(ISUB(NE.1) GO TO 27
17     I=INBOPT(N)
18     WRITE(6,990)N,INVACT(N)/IOTYS(N),ISRTOR(I),ITNBAG(I)NI
19     990     FORMAT(' ***FILLBO-NOOKUM',I5,' ONH',I5,' IOTYS=0.35,
20     &         I ISRTOR=',I5,' ITNBAG=',I5,' IPT=',I5)
21     27 CONTINUE
22     C     SET IPT EQUAL TO THE POWER NUMBER OF THE OLDEST, HIGHEST
23     C     PRIORITY REQUISITION FOR ITEM N ON BACKORDER STATUS.
24     IPT=INBOPT(N)
25     NSKU=EDESUB(IPT)
26     IOTYS=IOTYS(IPT)
27     ISRTOR=ISRTOR(IPT)
28     ITNBAG=ITNBAG(IPT)
29     IF(INVACT(N).LE.0) RETURN
30     C     *** NOTE ***
31     C     (A) THIS ROUTINE ASSUMES NO REQUISITIONS ARE FILLED USING
32     C     A FIFO, BY PRIORITY, ISRTOR RULE
33     C     (B) IF ONHAND ASSETS ARE INSUFFICIENT TO COMPLETELY FILL A
34     C     GIVEN REQUISITION, PARTIAL SHIPMENTS ARE INITIATED.
35     C     (C) REDUCED SHIPMENTS TO PARTIALLY FULFILL SEVERAL
36     C     OUTSTANDING BACKORDERS ARE NOT PERMITTED IN THIS ROUTINE
37     C
38     C     SET ITEST EQUAL TO THE REMAINING ON-HAND INVENTORY.
39     C     IF THE REQUISITION WAS TO BE COMPLETELY FILLED
40     C     ITNS=INVACT(N)-IOTYS
41     C     IS ITEST ABOVE THE SUPPORT LEVEL
42     IF(ITEST-GE.ISUL(N)) GO TO 60
43     C     IS THIS A PRIORITY 1 REQUISITION
44     IF(ISRTOR.NE.1) GO TO 60
45     C
46     C     PRIORITY 1 REQ
47     C
48     C     CAN THE REQ BE COMPLETELY SATISFIED FROM ON HAND STOCK
49     IF(INVACT(N)-IOTYS) GO TO 60
50     C     SHIP ALL REMAINING ONHAND STOCK IN PARTIAL FULFILLMENT
51     C     OF THIS REQUISITION
52     IOTYS=INVACT(N)

```

FILLBO

12 01 10-20-79 14.274 25

```

53      GO TO 85
54      C
55      C      PRIORITY IS BACKORDER
56      C
57      C      ARE ON-HAND ASSETS ABOVE THE SUPPORT LEVEL
58      60 IF (INVTCT(N) > LBL * ISUL(N))      GO TO 70
59      C      SHIP DOWN TO THE SUPPORT LEVEL
60      IOTIS = INVTCT(N) - ISUL(N)
61      C      UPDATE      STOCK STAGE RECORDS TO REFLECT THE HARTMAN SHIP
62      65      CONTINUE
63      CALL FILLST(N, IOTIS, IPRY, ITNO, 0, NKKV)
64      IOTIS(IPT) = IOTIS(IPT) - IOTIS
65      C      AT THIS POINT, NO FURTHER SHIPMENTS SHOULD BE POSSIBLE, SINCE
66      C      HIGH PRIORITY REQUISITIONS ARE PROCESSED FIRST
67      70 RETURN
68      C      SHIP TO FILL THE ENTIRE REQUISITION
69      80 IOTIS = IOTIS(IPT)
70      C      UPDATE      STOCK STAGE RECORDS
71      CALL FILLST(N, IOTIS, IPRY, ITNO, 0, NKKV)
72      C      REMOVE      REQUISITION FROM THE BACKORDER FILE
73      140 INOPT(N) = INACT(IPT)
74      NLOCKK, NROCBN = 1
75      NLOCKK, NROCBN = IPT
76      C      RETURN      TO BEGINNING OF      ROUTINE TO SEE IF ANY MOR
77      C      NEED TO BE PROCESSED.
78      GO TO 25
79      END

```

Subroutine: FILLST**Function:**

This routine updates backorder statistics to reflect shipment of IQTY units of SKU N to satisfy a backorder. If the backorder represents a requisition from some other stocking location, an appropriate receipt event is scheduled.

Calling Parameters:

- N = The Stock Keeping Unit which will provide the assets to be shipped.
- IQTY = The quantity of assets to be shipped
- IPRI = The priority of the backorder being filled.
- ITMBO = The time that the requisition was placed into a backorder status.
- NR = The number of requisitions to be removed from the backorder file by this shipping action. If NR = 1, the number of back-ordered requisitions is reduced by 1. Otherwise, NR = 0, reflecting a shipment to partially fill a current backorder.

NSKU = The SKU that originated the requisition.

Description:

This subroutine first computes the length of time that the backorder has been outstanding, and calls subroutine CUM to update the backorder-day statistics arrays (IBODAT and IBODAI).

FILLST then schedules appropriate Receive Parts events. If NSKU is greater than 1000, the backorder represents a requisition to provide parts for the repair of reparable generation number NSKU. In this case, subroutine FILLST schedules an LRU Receive Parts Event (Event Code 17) to occur 10 Time Measurement Units from the current time. If NSKU is less than 1000, but NSKU is not equal to N, current backorder represents a requisition to supply some other stocking location (e.g., a base or an overhaul facility) In this case, FILLST schedules a Receive Parts Event (Event Code 2).

Finally, subroutine FILLST updates records of on hand inventory for SKU N, and calls subroutine CUM to update the shipping statistics arrays ISHIPP and ISHIPI.

C 01 10-20-79 10.279

```

1      SUBROUTINE FILLST(N,IOTY,STIME,STIME0,NSKU)
2      C
3      C   UPDATE BACKORDER STATISTICS TO REFLECT SHIPMENT OF IOTY
4      C   UNITS OF ITEM N TO SATISFY A PRIORITY 1 BACKORDER.
5      C   IF NR=1, THE NUMBER OF REQUISITIONS BACKORDERED IS REDUCED BY 1
6      C   OTHERWISE, NR=0, REFLECTING A SHIPMENT TO PARTIALLY FILL
7      C   THE CURRENT BACKORDER.
8      C
9      C   IF NSKU.GE. 1000, SCHEDULE A RECEIVE PARTS
10     C   EVENT (EVENT CODE 479)
11     C
12     COMMON/ITIME/ITIME
13     COMMON/IWDAT/IWDAT
14     COMMON/IBODAT/IBODAT(1)
15     COMMON/IBODAT/IBODAT(1)
16     COMMON/LTPROD/LTPROD(1)
17     COMMON/ISNUG/ISNUG
18     COMMON/ISNPT/ISNPT(1)
19     COMMON/ISNPT/ISNPT(1)
20     COMMON/NSOTU/NSOTU(1)
21     COMMON/NSOTU/NSOTU(1)
22     COMMON/NSOTR/NSOTR(1)
23     COMMON/NSOTR/NSOTR(1)
24     COMMON/INVACT/INVACT(1)
25     C
26     C   UPDATE BACKORDER TIME STATISTICS
27     C
28     IBOTH=ITIME-IBH0
29     IUNDAT=IBOTH*IOTY
30     C
31     CALL CUN(IBODAT,IUNDAT,N)
32     IF(IPTIME.1)CALL CUN(IBODAT,IUNDAT,N)
33     C
34     C
35     C   IF NSKU > 1000, SCHEDULE A RECEIVE PARTS
36     C   EVENT (EVENT CODE 477).
37     C
38     JTIME= STIME + 10
39     IF(NSKU.GE.1000) CALL ENTER(STIME+17,N,IOTY,NSKU)
40     C
41     C   IF REQUISITION IS TO REFLECT A LOWER
42     C   SUPPLY LEVEL, SCHEDULE A RECEIVE EVENT.
43     C
44     JTIME= STIME + LTPROD(NSKU)*IOTY
45     IF((NSKU.LE.1000).AND.(N.NE.NSKU))
46     &   CALL ENTER(JTIME,1,NSKU,IOTY,N)
47     C
48     C   REDUCE ONHAND STOCKS AND BACKORDER STATUS
49     C
50     INVACT(N)=INVACT(N)+IOTY
51     NSOTU(N)=NSOTU(N)-IOTY
52     NSOTR(N)=NSOTR(N)-NR

```

FILLST

T 01 10-20-79 10.876

```

53      CALL CUN(IENIST,IOTY,N)
54      IF(IENR,NB,1)GO TO 20
55      NBOIN(N)=NBOIN(N)+IOTY
56      NBOIN(N)=NBOIN(N)-NR
57      CALL CUN(IENIST,IOTY,N)
58      20      CONTINUE
59      IF(IENR,NB,1)RETURN
60      WRITE(6,23)N,NBKU,IOTY,IENIST,NB,NBOTU(N),NBOIN(N)
61      2      NBOIN(N),NBOIN(N)
62      23 FORMAT(10X," ***FILLST N=IEN," NBKU=I,IS,
63      4      " IOTY=I,IS,
64      4      " IENR=I,IS," ITHBAC=I,IS," NR=I,IS,
65      4      " I20,NBOTU=I,IS," NBOIN=I,IS," NBOIN=I,IS,
66      4      " NBOIN=I,IS)
67      RETURN
68      END

```

Subroutine: INFEL

Function:

This routine initializes the Future Events List to an empty status.

Description:

Subroutine INFEL initializes the Future Events List. Subsequently, Subroutines ENTER and REMOVE update this list. See Volume I, Section II, for a detailed discussion of these activities.

```
1      SUBROUTINE INFEL
2      C      THIS ROUTINE INITIALIZES THE FUTURE EVENTS
3      C      LIST TO AN EMPTY STATUS
4      COMMON/NENTRY/NENTRY
5      COMMON/NFPMAX/NFPMAX
6      COMMON/NTIME/NTIME
7      COMMON/ILOCVE/ILOCVE(1)
8      NENTRY=0
9      NTIME=9999999
10     DO 1 I=1,NFPMAX
11     1 ILOCVE(I)=1
12     RETURN
13     END
```

INFEL

Subroutine: INGASP

Function:

This routine initializes the GASP file system.

Calling Parameters:

KTRY = The maximum number of entries into the GASP file system

KFILE = The number of files to be utilized

KSET = The dimension of NSET; that is, the maximum number of data elements to be stored in the GASP file system

KATR = The number of attributes of a file entry

Description:

Subroutine INGASP first sets the GASP COMMON variables NENTRY, NNFIL, NNSET, and NNATR to the respective calling parameters. The GASP time variable TNOW is then set to zero, and GASP file variables KKRNK (L) and IINN(L) are set so that all files will be ordered based upon ascending values of attribute 3. Finally, subroutine SET is called to initialize the GASP file system pointers.

04 10-20-78 10.287 .5

*SUBNAME:TIME/OBJ/INGASP.O(SCD,MOG0)

*INGASP,S

SUBROUTINE INGASP(KTRY,KFIL,KSET,KATR)

COMMON /GCOM1/ ATRIB(30),JEVNT,MFA,MFE(100),MLE(100),MSTOP,WCRD0,
NNADO,NNAPT,NNATR,NNPFL,NNQ(100),NNTRY,NPENT,PPARM(50,4),TNOW,TTREG,

TTCLR,TTFIN,TTIRB(30),TTSET

COMMON /GCOM0/ EENQ(100),TYNN(100),KERNK(100),MMAO(100),

QOTIN(100),SSOBV(25,5),SSTPV(25,6),VVNO(100)

COMMON/IDBUG/IDBUG

COMMON/ITIME/XTIME

MAXMIN=KTRY*(KATR+2)

IF(IDBUG.EQ.1)WRITE(6,13)ITIME,KTRY,KFIL,KSET,KATR,MAXMIN

13 FORMAT('*****INGASP=-XTIME=',I8,/T20,'MAX FILE ENTRIES',T50,I5/

T20,'NUMBER OF FILES',T50,I5/

T20,'DIMENSION OF NSET',T50,I5/

T20,'NO. OF ATTRIBUTES',T50,I5/

T20,'MAX FILE ENTRIES',T50,I5)

INITIALIZE GASP FILE SYSTEM

SET FILE SYSTEM PARAMETERS
FIRST,SET MAX FILE ENTRIES

6.50

NNTRY=KTRY

SET NUMBER OF FILE

NNFIL=KFIL

SET DEMINSION OF NSET

NNSET=KSET

SET NUMBER OF ATTRIBUTES

NNATR=KATR

NNAPT=NNATR+2

NNADO=NNATR+1

SET TIME PARAMETER

TNOW=0

SET DEFAULT FILE STRUCTURE BASED ON LOW VALUE
FIRST IN ATTRIBUTE 3

DO 50 IFL=1,NNFIL

KERNK(IFL)=3

INGASP

```

53      IINN(IPL)=1
54      SO CONTINUE
55      C
56      C      INITIALIZE FILE SYSTEM POINTERS
57      C
58      C      CALL SET
59      C
60      C**      END OF GASP INITIALIZATION**      *
61      RETURN
62      END

```

Subroutine: INITAL

Function:

This routine sets the value of RIME timing variables and initializes the Future Events List.

Description:

First, INITAL calls subroutine INFEL to initialize the Future Events List. Next, major parameters on the Backorder File are set, and the Backorder File is initialized.

RIME time parameters are then set assuming that there are 100 time units in each simulated day. At present, subroutine INITAL assumes that there are seven days per week, four weeks per month, three months per quarter, and four quarters in each year.

The routine then initializes the simulation clock and the statistics collection index; specifically, it sets:

ITIME = 0

ITIMZ = 1

INITAL then sets the other timing variables discussed in Section II, Volume I.

Finally, INITAL places events for event codes 6, 10, 11, and 20 on the Future Events List. If a Trace Event is to be utilized (i.e., if ISTRAC is greater than 0), and appropriate trace event (Event Code 13) is also scheduled.

01 10-20-79 10.298 JS

```

1  *#RUN=IRIME/OBS/INITAL.O.W/DCD.W0001
2  *INITAL.S
3  SUBROUTINE INITAL
4  C      THIS ROUTINE INITIALIZES THE FUTURE EVENTS LIST AND RECORDS
5  C      FILES. SETS THE TIMING PARAMETERS USED BY THIS SIMULATION,
6  C      AND SCHEDULES INITIAL ENDOGENOUS EVENTS.
7  C
8  COMMON/IDBUG/IDBUG
9  COMMON/ITRACE/ITRACE,ISTRAC
10 COMMON/IDDEV/IDDEV
11 COMMON/IDLEV/IDLEV
12 COMMON/IDSTAT/IDSTAT
13 COMMON/ITHQ/ITHQ
14 COMMON/IDEN/IDEN
15 COMMON/ITQTR/ITQTR
16 COMMON/IQTRN/IQTRN
17 COMMON/ISTOCK/ISTOCK,IDSTOC
18 COMMON/ISTOP/ISTOP
19 COMMON/ISTAT/ISTAT
20 COMMON/ITCANS/ITCANS,IDCANS
21 COMMON/ITDAY/ITDAY
22 COMMON/ITDIY/ITDIY
23 COMMON/ITFOR/ITFOR,IDFOR
24 COMMON/ITHO/ITHO
25 COMMON/ITIME/ITIME
26 COMMON/ITINV/ITINV
27 COMMON/ITLEV/ITLEV
28 COMMON/ITMTH/ITMTH
29 COMMON/ITQTR/ITQTR
30 COMMON/ITWEEK/ITWEEK
31 COMMON/ITYEAR/ITYEAR
32 COMMON/MBODAT/MBODAT
33 COMMON/MBMAX/MBMAX
34 COMMON/MBENTR/MBENTR
35 COMMON/MBEMAX/MBEMAX
36 COMMON/MBIRST/MBIRST
37 COMMON/MBTEN/MBTEN
38 COMMON/MBOC/MBOC
39 COMMON/MBOCBK/MBOCBK
40 COMMON/MBTIME/MBTIME
41 COMMON/ILOCBK/ILOCBK(200)
42 C
43 IF(IDBUG_EO.1)WRITE(6,113)
44 113 FORMAT('*****INITAL.....INITIALIZE PEL,GASP,DO-FILES.AND EVENTS')
45 C
46 C      INITIALIZE THE FUTURE EVENTS LIST
47 C
48 NPEMAX=500
49 CALL INPEL
50 C
51 C      INITIALIZE GASP FILE SYSTEM
52 C

```

INITAL

T 01 10-20-79 13.293 JS

53 CALL INSBSP(100.27600.4)

54 C

55 C

56 C

SET PARAMETERS FOR BACKORDER FILE

57 C

58

~~NRMAX=200~~

59

NLOCK=NRMAX

60

DO 10 I=1,NRMAX

61

10 NLOCK(I)=NRMAX+1-I

62 C

63 C

64 C

65 C

SET TIMING VARIABLES BASED ON 100 TIME UNITS PER DAY

66 C

7 DAYS/WEEK, 13 WEEKS/QUARTER, 4 QUARTERS/YEAR

67

ITDAY=100

68

ITWEEK=7*ITDAY

69

ITMTH=4*ITWEEK

70

ITOTR=3*ITMTH

71

ITYEAR=4*ITOTR

72 C

73 C

SET TIMING VARIABLES FOR MANAGEMENT AND DATA COLLECTION EVENTS

74 C

CURRENT SIMULATION CLOCK TIME

75

ITIME=0

76 C

CURRENT STATISTICS COLLECTION INTERVAL

77

ITINV=1

78 C

END OF CURRENT QUARTER

79

IOTRND=ITOTR

80 C

TIME OF FIRST HQ USAR BUDGET AUTHORIZATION

81

ITHQ=10

82 C

TIME BETWEEN HQ USAR BUDGET REVISIONS

83

IDITHQ=5*ITHQ

84 C

TIME OF FIRST DIVISION LEVEL REVIEW

85

ITDIV=20

86 C

TIME BETWEEN DIVISION LEVEL REVIEWS

87

IDDIV=ITMTH

88 C

TIME OF FIRST STOCK LEVEL COMPUTATION

89

ITLEV=10*ITOTR

90 C

TIME BETWEEN STOCK LEVEL COMPUTATIONS

91

IDLEV=ITOTR

92 C

TIME OF FIRST STOCK STATUS REVIEW

93

ISTOCK=40

94 C

TIME BETWEEN STOCK STATUS REVIEWS

95

IDSTOCK=2*ITWEEK

96 C

TIME TO ACTIVATE STATISTICS COLLECTION ROUTINE

97

ISTAT=ITWEEK*1

98 C

TIME BETWEEN STATISTICAL UPDATES

99

IDSTAT=ITWEEK

00 C

STOP AFTER SIMULATION INOTR QUARTERS

01

ISTOP=INOTR*ITOTR

02 C

03 C

PLACE INITIAL MANAGEMENT AND DATA COLLECTION EVENTS ON

04 C

THE FUTURE EVENTS LIST.

T 01 10-20-79 12.293 JS

```

05      C      STAT=REVIEW STOCK STATUS
06      C*****
07      C      LEVEL=COMPUTE INV CONTROL LEVELS
08      CALL ENTER(ITEVL,6,0,0,2DNEVL)
09      C
10      C      SCHEDULE INITIAL PROVISIONING EVENT
11      C
12      INPTIR=1
13      CALL ENTER(INPTIR,20,0,0,0)
14      C      SSTAT=ACCUMULATE STATUS STATISTICS
15      CALL ENTER(ISTAT,11,1,0,0)
16      C      SCHEDULE END OF SIMULATION RUN
17      CALL ENTER(ISTOP,10,0,0,0)
18      C
19      C      CREATE FORECAST EVENT
20      C
21      100 CONTINUE
22      IFOR=ITSTR
23      IDFOR=ITSTR
24      C      FORUPD=--UPDATE DEMAND HISTORY FILES
25      C*****
26      C      DEMPAR=--GENERATE DEMAND
27      C*****
28      C
29      C      CANCEL=--CANCELLATION REVIEW EVENT GOES HERE=
30      C
31      MBODAY=100
32      IDCANB=ITMNTN
33      C
34      C      SET TRACE EVENTS
35      C
36      IF(ISTRAC.LE.0) GO TO 200
37      CALL ENTER(ISTRAC,13,0,0,0)
38      CALL ENTER(ISTRAC,13,0,0,0)
39      200 CONTINUE
40      RETURN
41      END

```

1 **BUN=,TIME/OBJ/INITLB,0,N(BCD,NOGO)

39

2 *INITLB.S

3 SUBROUTINE INITAL

4 C THIS ROUTINE INITIALIZES THE FUTURE EVENTS LIST AND BACKORDER
5 C FILES, SETS THE TIMING PARAMETERS USED BY THIS SIMULATION,
6 C AND SCHEDULES INITIAL ENOGENOUS EVENTS.

7 C

8 COMMON/IDBUG/IDBUG

9 COMMON/ITRACE/ITRACE,ISTBAC

0 COMMON/IDDIV/IDDIV

1 COMMON/IDLEV/IDLEV

2 COMMON/IDSTAT/IDSTAT

3 COMMON/IDTNO/IDTNO

4 COMMON/IKDEM/IKDEM

5 COMMON/INQTR/INQTR

6 COMMON/IOTEND/IOTEND

7 COMMON/ISTOCK/ISTOCK,IDSTOC

8 COMMON/ISTOP/ISTOP

9 COMMON/ISTAT/ISTAT

0 COMMON/ITCANE/ITCANE,IDCANE

1 COMMON/ITDAY/ITDAY

2 COMMON/ITDIV/ITDIV

3 COMMON/ITFOR/ITFOR,IDFOR

4 COMMON/ITHQ/ITHQ

5 COMMON/ITIME/ITIME

6 COMMON/ITINV/ITINV

7 COMMON/ITLEV/ITLEV

8 COMMON/ITMTH/ITMTH

9 COMMON/ITQTR/ITQTR

0 COMMON/ITWEEK/ITWEEK

1 COMMON/ITYEAR/ITYEAR

2 COMMON/MBODAY/MBODAY

3 COMMON/NBMAX/NBMAX

4 COMMON/NENTRY/NENTRY

5 COMMON/NPFXMAX/NPFXMAX

6 COMMON/NFIRST/NFIRST

7 COMMON/NITEM/NITEM

8 COMMON/NLOC/NLOC

9 COMMON/NLOCBK/NLOCBK

0 COMMON/NTIME/NTIME

1 COMMON/ILOCBK/ILOCBK(200)

2 C

3 IF(IDBUG.EQ.1)WRITE(6,113)

4 113 FORMAT('*****INITAL.....INITIALIZE FEL,GASP,SO-FILES,AND EVENTS')

5 C

6 C INITIALIZE THE FUTURE EVENTS LIST

7 C

8 NPFXMAX=500

9 CALL INPEL

0 C

1 C INITIALIZE GASP FILE SYSTEM

2 C

INITLB

```

3      CALL INGASP(100,2,600,4)
4      C
5      C
6      C      SET PARAMETERS FOR BACKORDAR FILE
7      C
8      C      NRMAX=100
9      C      NLOCBK=NRMAX
10     C      DO 10 I=1,NRMAX
11     C      10 ILOCBK(I)=NRMAX+1-I
12     C
13     C
14     C
15     C      SET TIMING VARIABLES BASED ON 100 TIME UNITS PER DAY
16     C      7 DAYS/WEEK, 13 WEEKS/QUARTER, 4 QUARTERS/YEAR
17     C
18     C      ITDAY=100
19     C      ITWEEK=7*ITDAY
20     C      ITMTH=4*ITWEEK
21     C      ITQTR=3*ITMTH
22     C      IYEAM=4*ITQTR
23     C
24     C      SET TIMING VARIABLES FOR MANAGEMENT AND DATA COLLECTION EVENTS
25     C      CURRENT SIMULATION CLOCK TIME
26     C      ITIME=0
27     C      CURRENT STATISTICS COLLECTION INTERVAL
28     C      ITINV=1
29     C      END OF CURRENT QUARTER
30     C      IQTRND=ITQTR
31     C      TIME OF FIRST HQ USAF BUDGET AUTHORIZATION
32     C      ITHQ=10
33     C      TIME BETWEEN HQ USAF BUDGET REVISIONS
34     C      IDTHQ=3*ITQTR
35     C      TIME OF FIRST DIVISION LEVEL REVIEW
36     C      ITDIV=20
37     C      TIME BETWEEN DIVISION LEVEL REVIEWS
38     C      IDDIV=ITMTH
39     C      TIME OF FIRST STOCK LEVEL COMPUTATION
40     C      ITLEV=30*ITQTR
41     C      TIME BETWEEN STOCK LEVEL COMPUTATIONS
42     C      IDLEV=ITQTR
43     C      TIME OF FIRST STOCK STATUS REVIEW
44     C      ISTOCK=40
45     C      TIME BETWEEN STOCK STATUS REVIEWS
46     C      IDSTOC=2*ITWEEK
47     C      TIME TO ACTIVATE STATISTICS COLLECTION ROUTINE
48     C      ISTAT=ITWEEK-1
49     C      TIME BETWEEN STATISTICAL UPDATES
50     C      IDSTAT=ITWEEK
51     C      STOP AFTER SIMULATION INQTR QUARTERS
52     C      ISTOP=INQTR*ITQTR
53     C
54     C      PLACE INITIAL MANAGEMENT AND DATA COLLECTION EVENTS ON
55     C      THE FUTURE EVENTS LIST.

```

```

05      C      STAT--REVIEW STOCK STATUS
06      C*****
07      C      LEVEL--COMPUTE INV CONTROL LEVELS
08      CALL ENTER(ITLEV,6,0,0,IDLEV)
09      C
10      C      SCHEDULE INITIAL PROVISIONING EVENT
11      C
12      INPTIM=1
13      CALL ENTER(INPTIM,20,0,0,0)
14      C      SSTAT--ACCUMULATE STATUS STATISTICS
15      CALL ENTER(ISTAT,11,1,0,0)
16      C      SCHEDULE END OF SIMULATION RUN
17      CALL ENTER(ISTOP,10,0,0,0)
18      C
19      C      CREATE FORECAST EVENT
20      C
21      100 CONTINUE
22      ITFOR=ITOTR
23      IDFOR=ITOTR
24      C      FORUPD--UPDATE DEMAND HISTORY FILES
25      C*****
26      C      DEMPAR--GENERATE DEMAND
27      C*****
28      C
29      C      CANCLB--CANCELLATION REVIEW EVENT GOES HERE--
30      C
31      MBODAY=100
32      IDCANB=ITMNTB
33      C
34      C      SET TRACE EVENTS
35      C
36      IF(ISTRAC,LE,0) GO TO 200
37      CALL ENTER(ITRACE,18,0,0,0)
38      CALL ENTER(ISTRAC,19,0,0,0)
39      200 CONTINUE
40      RETURN
41      END

```

Subroutine: INITEM, INITM1, INITM2

Function:

This routine reads in data for a new LRU/SRU data set, and initializes associated inventory variables to zero.

Description

This routine has two entry points, INITM1 and INITM2. Entry point INITM1 is called to initiate the processing of a given LRU/SRU data set. Entry point INITM2 is called to initialize inventory variables associated with all SKUs at the beginning of each simulation replication.

When INITM1 is first called, it sets the record size for random file 11 to 11 words. This file serves as a work file for holding inventory levels computed by the Levels Computation Module as a preprocessing step. The routine then reads in data records from file 07 defining the characteristics of the current LRU/SRU data set. It then sets lead time variables for each Stock Keeping Unit number based on this input data. Finally, INITM1 reads levels data from file 09, and writes this data to the random work file 11. (During the simulation of this LRU/SRU data set, subroutine LEVEL reads file 11 at the beginning of each quarter to determine authorized stock levels.) INITM1 then returns to the calling program.

Entry point INITM2 is called to initialize inventory status variables at the beginning of each simulation replication. When the routine is called, it sets the inventory status variables INVACT(N), INVDUE(N), and INWIP(N) to zero. The backorder status variables NBOTU, NBOIU, NBOIR, and NBOTR are also set to zero. Finally, the backorder pointer NBOPT(N) is set to zero and logic returns to the calling program.

```

1  *#RUN=:TIME/OBJ/INITEM.O.W(BCD,NOGO)
2  *INITEM,S
3  SUBROUTINE INITEM
4  C      READ IN AND INITIALIZE DATA FOR A NEW LRU/SRU SET
5  C
6  CHARACTER ALC,FSN,UM,NOUN,MSTCD
7  COMMON/FSN/ALC,FSN(4),UM,NOON(2),MGTC(4),ION,IOR,IPPL,IPPR
8  COMMON/GSLF/GSLF
9  COMMON/IDBUG/IDBUG
0  COMMON/NBASES/NBASES
1  COMMON/NSRU/NSRU
2  COMMON/ITDAY /ITDAY
3  COMMON/ITMNT/ITMNT
4  COMMON/NITEM/NITEM
5  COMMON/NDEM/NDEM
6  COMMON/NDHIS/NDHIS
7  COMMON/INLU/INLU
8  COMMON/INTYPE/INTYPE
9  COMMON/LEBUG/LEBUG
0  COMMON/IBOP/IBOP(3),IBOP(3)
1  COMMON/IDEMND/IDEMND(1,24)
2  COMMON/RMREQS/RMREQS(1)
3  COMMON/INVACT/INVACT(1)
4  COMMON/IBRT/IBRT(1)
5  COMMON/IDRT/IDRT(1)
6  COMMON/IDORT/IDORT(1)
7  COMMON/NORDPT/NORDPT(1)
8  COMMON/NDEMAG/NDEMAG(1)
9  COMMON/NRETAC/NRETAC(1)
0  COMMON/NREQAC/NREQAC(1)
1  COMMON/NDEMND/NDEMND(1,24)
2  COMMON/NRETUR/NRETUR(1,24)
3  COMMON/NREQ/NREQ(1,24)
4  COMMON/NDENT/NDENT(1)
5  COMMON/INVDUE/INVDUE(1)
6  COMMON/INWIP/INWIP(1)
7  COMMON/NBOPT/NBOPT(1)
8  COMMON/NBOTU/NBOTU(1)
9  COMMON/NBOIU/NBOIU(1)
0  COMMON/NBOIR/NBOIR(1)
1  COMMON/NBOTR/NBOTR(1)
2  COMMON/REQSIZ/REQSIZ(1)
3  COMMON/REQHAB/REQHAB(1)
4  COMMON/LTPROD/LTPROD(1)
5  COMMON/LTADM/LTADM(1)
6  COMMON/UCOST/UCOST(1)
7  COMMON/ADR/ADR(1)
8  COMMON/ISUL/ISUL(1)
9  COMMON/IREQ /IREQ(1,24)
0  COMMON/IRETUR/IRETUR(1,24)
1  COMMON/IRL/IRL(1)
2  COMMON/ITL/ITL(1)

```

INITEM

```

COMMON/IROL/IROL(1)
COMMON/IRQTY/IRQTY(1)
COMMON/RMTBR /RMTBR(1)
COMMON/RMEAN/RMEAN(1)
COMMON/RTREND/RTREND(1)
COMMON/RNAD /RNAD(1)
COMMON/RRSUM/RRSUM(1)
COMMON/KNT /KNT(1)
COMMON/GROQ/GROQ(3)
COMMON/GBOOF/GBOOF(3)
COMMON/ICDFOR/ICDFOR

```

C

CHARACTER BPSN*15

C

SET NUMBER OF PERIODS OF DATA INPUT

C

IDPER=NDEM

C

C

C

ENTRY INITH1

C

N=0

NBP2=NBASES+2

10 CONTINUE

READ(7,END=9999)IC1,IC2,IC3,ISEQ,BPSN,COST

IF(IDBUG.EQ.1) WRITE(6,13)IC1,IC2,IC3,ISEQ,BPSN,COST

13 FORMAT(3I5,2X,"ISEQ=",I8," PSN=",A15," UCOST=",F10.2)

IF(IC1.LE.0) GO TO 40

N=N+1

C

C

FUDGE DATA FOR NOW

C

DO 50 NN=1,NBP2

N=(N-1)*NBP2 + NN

LTADM(N)=2

LTROD(N)=8

IBET(N)=10

IDET(N)=20

IDERT(N)=30

UCOST(N)=COST

50 CONTINUE

GO TO 10

C

C

C

SET NITEM AND NSRU

40 CONTINUE

NITEM=(NBASES+2)*N

NSRU=0

IF(N.GT.1)NSRU=N-1

IF(IDBUG.EQ.1)WRITE(6,113)NITEM,NSRU

113 FORMAT("*****NITEM. NITEM=",I5," NSRU=",I5)

INITM1

```

25      RETURN
26      C
27      C-----BEGIN ITEM INITIALIZATION LOOP-----
28      C
29      ENTRY INITH2
30      IF(IDBUG.EQ.1)WRITE(6,123)
31      123 FORMAT("BEGIN INITH2...ITEM INITIALIZATION")
32      C
33      DO 100 NN=1,NITEM
34      N=NN
35      C          INDICATE ITEM HAS NDHIS PERIODS OF DEMAND HISTORY
36      C
37      NDENT(N)=NDHIS
38      C
39      C          ZERO DEMAND HISTORY RECORDS
40      NRETAC(N)=0
41      NDEMAC(N)=0
42      NREQAC(N)=0
43      C          SET INVENTORY DUE=IN TO ZERO
44      70 INV DUE(N)=0
45      NORDPT(N)=0
46      INWIP(N)=0
47      C
48      C          SET INITIAL BACKORDER COUNTERS TO ZERO
49      C
50      NBOTU(N)=0
51      NBOIU(N)=0
52      NBOIR(N)=0
53      NBOTR(N)=0
54      NBOPT(N)=0
55      C
56      C          ESTABLISH BEGINNING INVENTORY LEVELS
57      C
58      INVACT(N)=0
59      100 CONTINUE
60      C
61      RETURN
62      C
63      C          IF END OF FILE IS READ, PRINT MESSAGE, SET NITEM=0,
64      C          AND THEN RETURN
65      9999 CONTINUE
66      NITEM=0
67      NSRU=0
68      WRITE(6,9993)NITEM,NSRU
69      9993 FORMAT(/////"READ END OF FILE OF THE FOLLOWING VALUES WERE SET".
70      &      //T20,"NITEM=",I8,"      NSRU=",I8////)
71      RETURN
72      END

```

T 01 10-20-79 15.090 CS

```

1  *SUB=INITM2/INITM2.0(BCD,NOBOY
2  *INITM2.0
3  SUBROUTINE INITM
4  C      USED IN AND INITIALIZES DATA FOR A NEW LRU/SRU SET
5  C
6  CHARACTER ALC,FSH,UN,NOU,NOCD
7  COMMON/FSH/ALC,FSH(4),UNITNUM(27),NOVCB(4),ION,ION,IPPA,IPPA
8  COMMON/NOU/NOU(4)
9  COMMON/NOU/NOU(4)
10 COMMON/NOU/NOU(4)
11 COMMON/NOU/NOU(4)
12 COMMON/NOU/NOU(4)
13 COMMON/NOU/NOU(4)
14 COMMON/NOU/NOU(4)
15 COMMON/NOU/NOU(4)
16 COMMON/NOU/NOU(4)
17 COMMON/NOU/NOU(4)
18 COMMON/NOU/NOU(4)
19 C-----
20 COMMON/NOU/NOU(4)
21 COMMON/NOU/NOU(4)
22 COMMON/NOU/NOU(4)
23 COMMON/NOU/NOU(4)
24 COMMON/NOU/NOU(4)
25 COMMON/NOU/NOU(4)
26 COMMON/NOU/NOU(4)
27 COMMON/NOU/NOU(4)
28 COMMON/NOU/NOU(4)
29 COMMON/NOU/NOU(4)
30 COMMON/NOU/NOU(4)
31 COMMON/NOU/NOU(4)
32 COMMON/NOU/NOU(4)
33 COMMON/NOU/NOU(4)
34 COMMON/NOU/NOU(4)
35 COMMON/NOU/NOU(4)
36 COMMON/NOU/NOU(4)
37 COMMON/NOU/NOU(4)
38 COMMON/NOU/NOU(4)
39 COMMON/NOU/NOU(4)
40 COMMON/NOU/NOU(4)
41 COMMON/NOU/NOU(4)
42 COMMON/NOU/NOU(4)
43 COMMON/NOU/NOU(4)
44 COMMON/NOU/NOU(4)
45 COMMON/NOU/NOU(4)
46 COMMON/NOU/NOU(4)
47 COMMON/NOU/NOU(4)
48 COMMON/NOU/NOU(4)
49 COMMON/NOU/NOU(4)
50 COMMON/NOU/NOU(4)
51 COMMON/NOU/NOU(4)
52 COMMON/NOU/NOU(4)

```

INITM2

[illegible]

T 01 10-20-79 19.498 28

```

05      C
06      C      SET OVERHAUL FACILITY PARAMETERS
07      C
08      MD=M+NBASR+2
09      LTADR(MD)=JBBORD
10      LTPROD(MD)=JGVSKP
11      ZBRT(MD)=JDOBT
12      EORT(MD)=JDOBT+JERT
13      EDORS(MD)=JDOBT
14      WCOSS(MD)=WCOBT
15      GO TO 10
16      C
17      C      SET NITER AND NSRU
18      C
19      40 CONTINUE
20      NITER=NBASR+2)*N
21      NSRU=0
22      IF(M.GT.1)NSRU=N-1
23      IF(ITERU.SQ.1)WRITE(6,13)NITER,NSRU
24      113 FORMAT('*****NITER. NITER=8,NS,7 NSRU=8,15)
25      IWORP=NSRU+1
26      C
27      C      READ LEVELS FOR QTRS 1,3,5,7,9 AND ALL GROUPS
28      C      FOR THIS LAMDA
29      C      AND WRITE TO WORK FILE 19.
30      C
31      IF(ITERU.SQ.1)WRITE(6,43)
32      DO 60 KQ=1,ITER,2
33      DO 55 I=1,ITER
34      READ(9,38,END=70)IDENT,KLAN,LEUGP,KQTR,NUMB,IOVSL,
35      &KD,KC,INSL(I),K=1,NBASR)
36      33 FORMAT('10,5I8,T31,I3,T48,10H37
37      IF(ITERU.SQ.1)WRITE(6,53)
38      4 IDENT,KLAN,LEUGP,KQTR,NUMB,IOVSL,
39      &KD,KC,INSL(I),K=1,NBASR)
40      C
41      43 FORMAT('9 *****NITER1 INPUT LEVELS= IDENT KLAN LEUGP KQTR NUMB",
42      9 IOVSL KQ KC INSL*172H'.:')
43      53 FORMAT('26X,8T6,8T3)
44      C
45      C      CHECK THAT INDEXES ARE AS EXPECTED
46      C
47      IF(KLAN.NB.KLAN) GO TO 399
48      IF(LEUGP.NB.NEUGP) GOTO 399
49      IF(KQTR.NB.KQ) GO TO 399
50      C
51      C      DATA CHECKS. WRITE ISSUES TO FILE 19
52      C      OUTPUT DUPLICATE LEVELS FOR QTRS KQ AND KQ+1
53      C
54      INDEX=16+(KQ-1)*IT
55      WRITE(11:INDEX)IOVSL,KD,KC,NBOL
56      INDEX=16+KQ * IT

```

```

157      WRITE(11,'INDEX')IOVSL,KD,KE,YBBL
158      55 CONTINUE
159      60 CONTINUE
160      C
161      70 CONTINUE
162      RETURN
163      C
164      C      LEVEL RECORD PARAMETER WAS NOT AS EXPECTED.
165      C      STOP RUN.
166      399 CONTINUE
167      WRITE(6,68) '*****INITH1--LEVELS PARAMETERS DON'T MATCH**'
168      &      P HLAN="HLAN," NURBUS="NROUPE," KOF="EQ
169      WRITE(6,68) '*****LEVELS REAS ARE...'
170      WRITE(6,68)
171      &      IDENT,KLAN,LEUCP,KOTR,NUMBETOVEL,
172      &      AND,KC,BIBSL(N),K=1,NBASES)
173      63 FORMAT(IV)
174      WRITE(6,68) '*****INITH2--STOP RUN.'
175      STOP
176      C
177      C-----S-----BEGIN ITEM INITIALIZATION LOOP-----
178      C
179      ENTRY INITH1
180      IF(IEBUS.EQ.1)WRITE(6,128)
181      123 FORMAT('BEGIN INITH2...ITEM INITIALIZATION')
182      C
183      C
184      DO 100 N=1,NITEM
185      NTHN
186      C      SET INVENTORY DUE-IN TO ZERO
187      INVDUEIN=0
188      INWIP(N)=0
189      C
190      C      SET INITIAL BACKORDER COUNTERS TO ZERO
191      C
192      NBOIN(N)=0
193      NBOURN(N)=0
194      NBOZR(N)=0
195      NBOGR(N)=0
196      NBOPT(N)=0
197      C
198      C      ESTABLISH BEGINNING INVENTORY LEVELS
199      C
200      INVACT(N)=0
201      100 CONTINUE
202      C
203      IF(IEBUS.EQ.1)WRITE(6,63) '*****EXIT INITH2'
204      RETURN
205      C
206      C      IF END OF FILE IS REACHED PRINT MESSAGE, SET NITEM=0,
207      C      AND THEN RETURN
208      9999 CONTINUE

```

IT 01 10-20-79 14.698 63

```
209      WITEM=0
210      NSRU=0
211      WRITE(6,9993)WITEM,NSRU
212      9993 FORMAT(//////READ END OF FILE 07.THE FOLLOWING VALUES WERE SET",
213      &      //T80,"WITEM=",I8,"      NSRU=",I8//1/7
214      RETURN
215      END
**W      7 MEMORY EXPANDED. USE $LIMITS OR $ORIG= OPTION FOR NEXT RUN
```

Subroutine: ITRSLT

Function:

This routine records selected performance statistics for the current replication, and prints and punches results.

Description:

If the current group being simulated equals NFGRP, and if this is the first replication for the group, the statistics array JMEAS is set to zero. The routine then outputs 10 selected performance measures for the current LRU/SRU group. The 10 statistics recorded are depot buy dollars, depot backorder-days, base fills, base requisitions, and base backorder-days for LRU and SRUs, respectively. Summaries for both the 8th quarter of the simulation and for the 16th quarter of the simulation are then printed and punched.

01 10-20-79 10.308 JS

```

1  *#RUV=IRIME/888/ITRSL2.0(BCD,N086)
2  *ITRSL2.8
3  SUBROUTINE ITRSLT
4  COMMON/IZENT/ITNET,IOUT,IGRAPH,ISUMRY
5  COMMON/IDENT/IDENT
6  COMMON/MEAN/MEAN,MLAN
7  COMMON/MREPL/MREPL,MREPL
8  COMMON/MGROUP/MGROUP
9  COMMON/MFGP/MFGP
10 COMMON/INOTR/INOTR
11 COMMON/ITIEV/ITIEV
12 COMMON/IORDER/IORDER(16,3,6)
13 COMMON/IBAKDI/IBAKDI(16,3,6)
14 COMMON/IFILLI/IFILLI(16,3,6)
15 COMMON/IFILLT/IFILLT(16,3,6)
16 COMMON/IREOI/IREOI(16,3,6)
17 COMMON/IREOT/IREOT(16,3,6)
18 COMMON/IBODAI/IBODAI(16,3,6)
19 COMMON/IBODAT/IBODAT(16,3,6)
20
21 C
22 DIMENSION IMEAS(16,10),JMEAS(16,10)
23 DIMENSION ITOTL(10)
24 DATA NMEAS/10/
25
26 C
27 C      IMEAS(I,J) = VALUE OF MEASURE J FOR QTR I OF THIS
28 C                      REPLICATION.
29 C      JMEAS(I,J) = CUMULATIVE VALUE OF MEASURE J FOR QTR I
30 C                      FOR ALL REPLICATIONS UP TO AND INCLUDING
31 C                      THIS ONE.
32 C      NMEAS = 10 = NO. OF MEASURES RECORDED BY THIS ROUTINE.
33
34 C      ON FIRST REPLICATION, ZERO THE JMEAS-ARRAY
35 C      WHEN THE FIRST ITEM IS PROCESSED.
36
37 C
38 IF(MGROUP.EQ.MFGP) GO TO 20
39 IF(MREPL.EQ.1) GO TO 20
40 DO 10 I=1,16
41 DO 10 J=1,NMEAS
42   JMEAS(I,J)=0
43 10 CONTINUE
44 20 CONTINUE
45
46 C
47 C      COMPUTE STATISTICS FOR THIS REPLICATION
48 C
49 C
50 DO 50 I=1,INOTR
51   IMEAS(I,1)=IORDER(I,3,3)*JMEAS(I,1)
52   IMEAS(I,2)=IORDER(I,3,4)*JMEAS(I,2)
53   IMEAS(I,3)=IBODAT(I,2,3)/100 *JMEAS(I,3)
54   IMEAS(I,4)=IBODAT(I,2,4)/100 *JMEAS(I,4)
55   IMEAS(I,5)=IFILLT(I,2,1)*JMEAS(I,5)
56   IMEAS(I,6)=IFILLT(I,2,2)*JMEAS(I,6)
57   IMEAS(I,7)=IREOT(I,2,1)*JMEAS(I,7)

```

ITRSLT

01 10-20-79 13.508 5

```

3      IMEAS(I,8)=INQTR(I,2,2)-JMEAS(I,8)
4      C
5      IMEAS(I,9)=IBODAT(I,2,1)/100 -JMEAS(I,9)
6      IMEAS(I,10)=IBODAT(I,2,2)/100 -JMEAS(I,10)
7      C
8      C
9      50 CONTINUE
10     C
11     C
12     C      RECORD CURRENT VALUES FOR USE ON THE NEXT REPLICATION
13     C
14     DO 300 I=1,INQTR
15     DO 290 J=1,NMEAS
16     JMEAS(I,J)=JMEAS(I,J)+IMEAS(I,J)
17     290 CONTINUE
18     300 CONTINUE
19     C
20     C
21     C      OUTPUT RESULTS FOR THIS REPLICATION
22     C
23     C
24     WRITE(6,1)NGROUP,IDENT,HLAM,NREPL
25     1  FORMAT(1H1," *****ITRSLT...  NGROUP=",I2,I2,"  IDENT=",I2,I2,
26     &      ",  HLAM=",I2,I2,"  NREPL =",I2,I2)
27     WRITE(6,2)
28     C
29     IF((HLAM.GT.1).OR.(NREPL.GT.1))GO TO 180
30     KNT=8
31     IF(INQTR.GT.KNT)KNT=INQTR
32     WRITE(15,3)KNT,NGROUP,IDENT
33     IF(INQTR.GT.8)WRITE(16,3)INQTR,NGROUP,IDENT
34     3  FORMAT(4H1///T30,I2,"-QTR TOTALS"/T25,"NGROUP =",I3,
35     &      "  IDENT =",I3)
36     WRITE(15,2)
37     IF(INQTR.GT.8)WRITE(16,2)
38     2  FORMAT(//T20,"DEBOT",T40,"DEBOR",T60,3("BASE",16X)/
39     &      T17,"TORDER=",T39,"IBODAT",T58,"IFILLT",T79,"IREOT",
40     &      T99,"IBODAT"/
41     &      T8,"I1",T15,3("LNU      BRD      "/)
42     C
43     C
44     C
45     180 CONTINUE
46     DO 200 I=1,INQTR
47     IF(ITRNT.EQ.2) PUNCH 5,NRGRUP,IDENT,HLAM,NREPL,I,
48     &      (IMEAS(I,J),J=1,NMEAS)
49     5  FORMAT(5I2,2I10,8I6,"II")
50     WRITE(6,193)I,(IMEAS(I,J),J=1,NMEAS)
51     193 FORMAT(1H  .I5,10X10)
52     200 CONTINUE
53     C      COMPUTE 5 AND 16 QTR TOTALS
54     C

```

```

5      INDEX=0
6      IF(INOTR.EQ.1) INDEX=INDEX+INOTR
7      DO 320 J=1,NHEAS
8          ITOTL(J)=0
9          DO 310 I=1,INDEX
10             ITOTL(J)=ITOTL(J)+IMEAS(I,J)
11      310 CONTINUE
12      320 CONTINUE
13      C
14      C          IF ITWRT=1, PUNCH 8-TH SUMMARIES
15      C
16      IF(ITWRT.EQ.1) PUNCH 15,NGROUPEIDENT,MLAN,MREPL,INDEX,
17      &      (ITOTL(J),J=1,NHEAS)
18      15 FORMAT(5I2,2I10,8I6,"AA")
19      WRITE(8,333)INDEX,(ITOTL(J),J=1,NHEAS)
20      333 FORMAT(//1X,I2,"QTR TOTALS0/0X,10I10)
21      C
22      IF(MREPL.EQ.1)WRITE(15,343)
23      343 FORMAT(//)
24      WRITE(15,353)MLAN,MREPL,(ITOTL(J),J=1,NHEAS)
25      353 FORMAT(2E3,10I10)
26      C
27      IF(INOTR.EQ.0) GO TO 400
28      DO 360 J=1,NHEAS
29      DO 350 I=1,INOTR
30          ITOTL(J)=ITOTL(J)+IMEAS(I,J)
31      350 CONTINUE
32      360 CONTINUE
33      C
34      C          IF ITWRT=1, PUNCH 16-TH SUMMARIES
35      C
36      IF(ITWRT.EQ.1) PUNCH 35,NGROUPEIDENT,MLAN,MREPL,INOTR,
37      &      (ITOTL(J),J=1,NHEAS)
38      35 FORMAT(5I2,2I10,8I6,"MM")
39      WRITE(6,383)INOTR,(ITOTL(J),J=1,NHEAS)
40      C
41      IF(MREPL.EQ.1)WRITE(16,343)
42      WRITE(16,353)MLAN,MREPL,(ITOTL(J),J=1,NHEAS)
43      C
44      400 CONTINUE
45      RETURN
46      END

```

Subroutine: KNSKU

Function:

This function computes the statistics aggregation index K associated with Stock Keeping Unit number N.

Description:

The variable NSTLOC denotes the total number of stocking locations for each Federal Stock Number being simulated. This value is then used to determine the appropriate aggregation index associated with SKU N.

```
1 *****KNSKU.S
2     FUNCTION KNSKU(N)
3     C
4     C     THIS FUNCTION COMPUTES THE STATISTICS INDEX K
5     C     ASSOCIATED WITH KNSKU=N.
6     C     COMMON/NBASSES/NBSEES
7     NSTLOC=NBASSES+2
8     NDEX=MOD(N,NSTLOC)
9     IF(N.LE.NSTLOC) GO TO 10
10    C
11    C     N IS AN SRH
12    C
13    KNSKU=2
14    IF(NDEX.EQ.0) KNSKU=6
15    IF(NDEX.EQ.1) KNSKU=4
16    RETURN
17    C
18    C     N IS AN LRU
19    C
20    10 KNSKU=1
21    IF(N.EQ.1) KNSKU=3
22    IF(NDEX.EQ.0) KNSKU=5
23    RETURN
24    END
```

KNSKU .

Subroutine: LEVEL

Function:

This routine obtains levels data from random file 11, and then sets reorder, retention, termination, and support levels for each SKU being simulated.

Calling Parameters:

NN = Item Flag. If NN = 0, subroutine LEVEL computes levels for all items. Otherwise, the routine computes levels for item NN only.

Description:

Subroutine LEVEL is called to represent a stock level computation event. Stock level data is originally input to the RIME simulation model through entry point INITM1, and written to the random work file 11. When routine LEVEL is called, it obtains stock levels for the current quarter by reading the next (NSRU + 1) sets of levels records from file 11. It then computes the stock keeping unit number associated with each stocking location, and sets the reorder level IROL(N) to the values obtained from File 11. Subroutine LEVEL then sets the retention and termination levels (ITL (N) and IRL (N)) to very large numbers. As a result, termination and disposal actions never occur. In addition, the support level ISUL (N) is set to zero for all items.

```

1  *NRUN=TIME/OBJ/LEVEL,0(RCD,NOGO)
2  *LEVEL,S  COMPUTE CONTROL LEVELS
3  SUBROUTINE LEVEL(MN)
4  C
5  C      THIS ROUTINE COMPUTES      REORDER,STOCK OBJECTIVE,RETENTION,
6  C      TERMINATION, AND SUPPORT LEVELS.
7  C
8      COMMON/INBUG/INBUG
9      COMMON/COSHET/COSHET
0      COMMON/COSMLE/COSMLE
1  *   COMMON/COSORD/COSORD(3)
2      COMMON/CSTBRK/CSTBRK
3      COMMON/GSULF/GSULF
4      COMMON/GRIF/GRIF
5      COMMON/GSIF/GSIF
6      COMMON/GTLF/GTLF
7      COMMON/ITLEV1/ITLEV1
8      COMMON/IDLEV1/IDLEV1
9      COMMON/NITEM/NITEM
0      COMMON/NBASES/NBASES
1      COMMON/POLICY/ICDEEQ,ICDSL,EDOMAX,EQOMIN,SLMAX,SLMIN,RLF,TLF,SULF
2      COMMON/ICDSL/ICDSL
3      COMMON/ADR/ADR(1)
4      COMMON/LTADM/LTADM(1)
5      COMMON/LTPROD/LTPROD(1)
6      COMMON/IQTY/IQTY(1)
7      COMMON/IRL/IRL(1)
8      COMMON/IROL/IROL(1)
9      COMMON/ISUL/ISUL(1)
0      COMMON/ITL/ITL(1)
1      COMMON/RSIGLT/RSIGLT(1)
2      COMMON/REQSIZ/REQSIZ(1)
3      COMMON/UCOST/UCOST(1)
4      DATA Z/1.0/
5      DATA ITLMIN,IRLMIN/99999999.99999999/
6  C
7  C      COMPUTE LEVELS FOR ITEM MN.  IF MN=0,
8  C      COMPUTE LEVELS FOR ALL ITEMS,
9  C      THAT IS, ITEMS 1,2,...,NITEM
0  C
1      NLOC=NBASES+2
2      NF=NN
3      NI=NN
4      IF(MN.EQ.0)NF=1
5      IF(MN.EQ.0)NI=NITEM
6  C
7      DO 3000 NNN=NF,NL
8          N=NNN
9  C
10     C
11     C      COMPUTE PLANNING FACTORS
12     C

```

LEVEL

01 10=20=79 10.539 S COMPUTE CONTROL LEVELS

```

3      AMDA=COSHRT
4      RLTM=LTADM(N)+LTPROD(N)
5      IF(RLTM.LT.1.)RLTM=0.5
6      ****
7      C      FOR NOW, SET ANNUAL DEMAND=12
8      C      FOR THE OVERHAUL FACILITY, SET AD=24.
9      AD=12.
10     KMOD=MOD(N,NLOC)
11     IF(KMOD.EQ.0) AD=24.
12     C
13     C      NOTE...LEADTIME IS IS DAYS
14     C
15     RL=AD*RLTM/365.
16     RMR=AD/12.
17     UC=UCOST(N)
18     ADDR=UC*AD
19     C
20     C
21     C      FOR NOW, SET Q=1
22     C
23     Q=1
24     C
25     C      FOR NOW, SET SL=0
26     C      SL=0
27     C
28     C
29     C      CHECK      ORDER SIZE LIMITS
30     200 CONTINUE
31     EMX=EOQMAX*RMR
32     IF(Q.GT.EMX) Q=EMX
33     EMX=EOQMIN*RMR
34     IF(Q.LT.EMX) Q=EMX
35     IF(Q.LT.1.)Q=1.
36     C
37     C
38     C      LIMITS BASED ON MONTHS OF SUPPLY
39     C
40     520 CONTINUE
41     C      LOWER LIMIT
42     SLM=SLMIN*RMR
43     IF(SL.LT.SLM)SL=SLM
44     C      UPPER LIMIT
45     SLM=SLMAX*RMR
46     IF(SL.GT.SLM)SL=SLM
47     C
48     C
49     GO TO 600
50     C*****
51     C
52     600 CONTINUE
53     C
54     C      COMPUTE LEVELS

```

1 01 40-20-79 10,539 S COMPUTE CONTROL LEVELS

```
15      C
16      IROTY(N)=Q+0.5
17      IROL(N)=SL+RLT+0.5
18      ITL(N)=SLMAX*RMR+RLT+GTLF*RMR+0.5
19      IRL(N)=FLOAT(ITL(N))+GSLF*RMR+0.5
20      IF(ITL(N).LE.ITLMIN) ITL(N)=ITLMIN
21      IF(IRL(N).LE.IRLMIN) IRL(N)=IRLMIN
22      ISUL(N)=GSULF*RMR
23      C
24      C
25      IF(IHBUG.NE.1) GO TO 2990
26      WRITE(6,8903)N,IROTY(N),IROL(N),ITL(N),IRL(N),ISUL(N)
27      8903      FORMAT(4X,'***LEVELN=-N=',I5,' IROTY=',I5,' IROL=',I5,
28      &      ' ITL=',I5,' IRL=',I5,' ISUL=',I5)
29      2990 CONTINUE
30      3000 CONTINUE
31      RETURN
32      END
```

T 01 10-20-79 10.501 JS

```

1  *#RUN=IRIME/888/LEVEL2.0(BCD,N000)
2  *LEVEL2.8
3      SUBROUTINE LEVEL(MN)
4      C      THIS ROUTINE COMPUTES LEVELS FOR ALL
5      C      (NSRU+1) STOCK KEEPING UNITS BY READING THE NEXT (NSRU+1)
6      C      SETS OF LEVELS RECORDS ON FILE #1. IT THEN COMPUTES
7      C      THE STOCK KEEPING UNIT NUMBER ASSOCIATED WITH EACH
8      C      NSN, AND SETS THE APPROPRIATE COMMON VARIABLES.
9      C
10     COMMON/INBUS/INBUS
11     COMMON/ITLEV/ITLEV
12     COMMON/IDLEV/IDLEV
13     COMMON/NSRU/NSRU
14     COMMON/NETEN/NETEN
15     COMMON/NSSES/NSSES
16     COMMON/LTADN/LTADN(1)
17     COMMON/LTPROD/LTPROD(1)
18     COMMON/IRQTY/IRQTY(1)
19     COMMON/ISL/ISL(1)
20     COMMON/IROL/IROL(1)
21     COMMON/ISUL/ISUL(1)
22     COMMON/ITL/ITL(1)
23     COMMON/ITINV/ITINV
24     C
25     C
26     DIMENSION IBSL(8)
27     C
28     KO=ITINV
29     ITGRP=NSRU+1
30     NLOC=NSSES+2
31     DO 300 I=1,ITGRP
32     C
33     C      INPUT LEVELS
34     INDEX=16*(KO-1)+I
35     READ(1,INDEX)IOVSL,KD,KC,IBSL
36     IF(INBUS.EQ.1)WRITE(6,33)KO,KD,KC,IBSL
37     33 FORMAT('***LEVEL2--QTR="I3," Y="I3," IOVSL="I3,'
38     &      ' KD="I3," KC="I3," IBSL="I3')
39     C
40     C
41     C      COMPUTE DEPOT AND OVERHAUL SKU NUMBERS
42     C
43     NSKUDP=(I-1)*NLOC+1
44     NSKUOV=NSKUDP+NLOC-1
45     C
46     C      SET RECORDER LEVELS FOR DEPOT AND OVERHAUL FACILITIES
47     C
48     IROL(NSKUDP)=KD+KC
49     IROL(NSKUOV)=IOVSL
50     C
51     C      SET BASE LEVELS
52     C

```

LEVEL 2

T 01 10-20-79 10.501 JS

```

53      DO 50 N=1,NBASES
54      NSKU=NSKUDF+NB
55      ISOL(NSKU)=ISOL(NB)
56      50 CONTINUE
57      C
58      C          SET OTHER LEVELS TO CONSTANT VALUES
59      C
60      DO 80 M=NSKUDF,NSKUOV
61      IROTY(M)=1
62      IRL(M)=9999999
63      IRL(M)=9999999
64      ISUL(M)=0
65      80 CONTINUE
66      C
67      C
68      IF(INBUB.EQ.1) GO TO 2990
69      DO 90 M=NSKUDF,NSKUOV
70      WRITE(6,8903)M,IROTY(M),IRL(M),ITL(M),IRL(M),ISUL(M)
71      8903      FORMAT(4X,'***LEVELS28-M=',I5,' IROTY=',I5,' IRL=4.33,
72      &          ' IRL=',I5,' IRL='7I5.4 ISUL=',I5)
73      90 CONTINUE
74      2990 CONTINUE
75      3000 CONTINUE
76      RETURN
77      END

```

Subroutine: NRTS

Function:

Event Code 19. This routine records a NRTS event for SKU N, and updates base and depot Work-in-Process records accordingly.

Calling Parameters:

N = The Stock Keeping Unit number of the inventory location which is originating the NRTS action.

IQTY = The number of units to be NRTS.

NJOB = The reparable generation number which is the source of the NRTS assets

Description:

Subroutine NRTS first computes the Stock Keeping Unit number of the depot associated with stocking location N. It then increases the work-in-process inventory for the depot SKU, and decreases the work-in-process for stocking location N. Subroutine CUM is then called to update the INRTS statistics arrays for both the depot and base locations. Logic then returns to the calling program.

```

1      *NRTS,S
2      SUBROUTINE NRTS(N,IQTY,NJOB)
3      C
4      C      EVENT CODE 19.
5      C      THIS EVEN RECORDS A NRTS FOR SKU N.
6      C      BASE WIP IS THEN DECREASED, AN
7      C      DEPOT WIP IS CORRESPONDINGLY, INCREASED
8      C
9      COMMON/IDBUG/IDBUG
10     COMMON/INWIP/INWIP(1)
11     COMMON/INRTS/INRTS(10,3,6)
12     COMMON/NBASES/NBASES
13     C
14     C      DETERMINE LRU/SRU NUMBER.
15     C      LRU=0,SRU NO 1=1,ETC.
16     C
17     NLOC=NBASES+2
18     NSR=(N-1)/NLOC
19     C
20     C      COMPUTE DEPOT STOCK KEEPING UNIT
21     C
22     NDSKU=(NSR)*NLOC+1
23     C
24     C      UPDATE WIP AT DEPOT AND BASE
25     C
26     INWIP(N)=INWIP(N)-IQTY
27     INWIP(NDSKU)=INWIP(NDSKU)+IQTY
28     C
29     C      INCREMENT NRTS COUNTERS FOR BOTH BASE AND DEPOT
30     C
31     CALL CUM(INRTS,IQTY,N)
32     CALL CUM(INRTS,IQTY,NDSKU)
33     C
34     C
35     IF(IDBUG.EQ.1)WRITE(6,113)N,IQTY,NJOB,NDSKU,INWIP(N),
36     & INWIP(NDSKU)
37     113 FORMAT("*****NRTS=-N=",I5," QTY=",I5," NJOB=",I5,
38     & " NDSKU=",I5," WIP(N)=",I5," WIP(NDSKU)=",I5)
39     C
40     C
41     C
42     RETURN
43     END

```

NRTS

Subroutine: ORDER, ORDERV

Function:

This routine updates statistics to reflect an order for IBQ units of SKU N, and schedules associated receipt transactions. Entry point ORDERV is called to represent shipments of initial provisioning assets.

Calling Parameters:

N = The SKU originating the requisition

IBQ = The quantity of assets requisitioned

JTIME = The time that the stock associated with this order is scheduled to be received. JTIME is computed within subroutine ORDER and returned to the calling program.

Description:

This routine reflects an order for IBQ units of SKU N, and places appropriate stock receipt transactions on the Future Events List. The variable IPHASE indicates whether the current order is associated with an initial provisioning or a replenishment action. IPHASE = 1 denotes an initial provisioning order, while IPHASE = 2 denotes a replenishment order. In an initial provisioning action, assets

are procured by the depot and shipped directly and immediately to the requisitioning location. For replenishment actions, however, the exact activities taken depends upon whether or not the stocking location under consideration is a depot. If the inventory location N submitting the order represents a base or an aircraft overhaul facility, subroutine ORDER schedules a replenishment requisition (Event Type 1) event to occur at the depot LTADM (N) days in the future. This action is taken since both the bases and the aircraft overhaul facility are resupplied from the depot. On the other hand, if the originating stocking location N is a depot, an outside vendor is the source of resupply. Consequently, subroutine ORDER schedules a stock receipt event (Event Type 2) for IBQ units to occur a leadtime in the future. The total leadtime consist of the sum of administrative leadtime (LTADM(N)) and production leadtime (LPPROD(N)). Finally, subroutine CUM is called to update the performance statistics IORDER, ILGORD, and ISMORD.

```

1  *#RUN=:TIME/OBJ/ORDER.O(BCD,NOCO)
2  *ORDER.S
3  SUBROUTINE ORDER(N,IBO,JTIME)
4  C      THIS ROUTINE UPDATES STATISTICS TO REFLECT AN ORDER FOR IBO
5  C      UNITS OF ITEM N, WITH DELIVERY DATE SCHEDULED FOR JTIME, AND
6  C      PLACES THE ASSOCIATED RECEIPT TRANSACTION ON THE FUTURE E
7  C      LIST.
8  C
9  C      IPHASE = 2 DENOTES REPLENISHMENT ORDER
10 C
11 C      IPHASE = 1 DENOTES INITIAL PROVISIONING
12 C
13 C      IN INITIAL PROVISIONING, ASSETS ARE PROCURED BY THE DEPOT
14 C      BUT SHIPPED DIRECTLY AND IMMEDIATELY TO THE REQUISITIONING
15 C      LOCATION.
16 COMMON/IDBUG/IDBUG
17 COMMON/IOBLIG/IOBLIG
18 COMMON/NLOC/NLOC
19 COMMON/NBASES/NBASES
20 COMMON/CSTBRK/CSTBRK
21 COMMON/ISHORD/ISHORD(16,3,6)
22 COMMON/ILGORD/ILGORD(16,3,6)
23 COMMON/IORDER/IORDER(16,3,6)
24 COMMON/INVDUE/INVDUE(1)
25 COMMON/LTADM/LTADM(1)
26 COMMON/LTPROD/LTPROD(1)
27 C
28 COMMON/UCOST/UCOST(1)
29 COMMON/JPRIOR/JPRIOR(500)
30 COMMON/ITIME/ITIME
31 COMMON/ITDAY/ITDAY
32 COMMON/IBOPCT/IBOPCT(3),IBOPLG(3)
33 C
34 C      SET REPLENISHMENT FLAG
35 C
36 IPHASE=2
37 C
38 C      DETERMINE STOCKING LOCATION FOR SKU N
39 C
40 NBP2=NBASES+2
41 LOC=MOD(N,NBP2)
42 C
43 C      IF N IS A DEPOT LOCATION, LOC=1
44 C
45 IF(LOC.EQ.1) GO TO 20
46 C
47 C      LOCATION IS NOT THE DEPOT
48 C
49 JTYPE=1
50 NS=(N-1)/NBP2
51 IP3=NS*NBP2 + 1
52 IP5=100*N + 2

```

ORDER

12 04 10-20-79 10.536 S

```

83      JTIME=JTIME + ITDAY*(LTADM(N))
84      GO TO 30
85      C
86      ENTRY ORDERV(N,IBQ,JTIME)
87      C      ORDER DIRECT FROM VENDOR IBQ UNITS OF ITEM N,
88      C      FOR DELIVERY AT TIME=JTIME
89      C
90      C      COMPUTE SKU OF DEPOT = NDEP
91      C
92      NBP2= NBASES + 2
93      NS=(N-1)/NBP2
94      NDEP=NS*NBP2 + 1
95      C
96      C      SET FLAGS FOR INITIAL PROVISIONING
97      C
98      10 CONTINUE
99      IPHASE=1
100     JTYPE=2
101     IP3=N
102     IP4=IBQ
103     IP5=0
104     GO TO 30
105     C
106     C      LOCATION IS A DEPOT
107     C
108     20 CONTINUE
109     JTIME=JTIME + ITDAY*(LTADM(N)+LTPROD(N))
110     JTYPE=2
111     IP3=N
112     IP5=0
113     30 CONTINUE
114     C
115     C
116     C      PLACE ORDER FOR IBQ UNITS OF ITEM IP3
117     C
118     CALL ENTER(JTIME,JTYPE,IP3,IBQ,IP5)
119     C      UPDATE      STATUS STATISTICS
120     IOBLIG=FLOAT(IOBLIG)+FLOAT(IBQ)*UCOST(N)
121     INVDUE(N)=INVDUE(N)+IBQ
122     DVORD=FLOAT(IBQ)*UCOST(N)
123     IF(JTIME.LE. 0) GO TO 100
124     IF(IPHASE.EQ.1)CALL GUM(IORDER,IBQ,NDEP)
125     IF(IPHASE.EQ.2)CALL GUM(IORDER,IBQ,N)
126     DVORD=FLOAT(IBQ)*UCOST(N)
127     IF(DVORD.GE. CSTBRK) CALL GUM(XLGORD,IBQ,N)
128     IF(DVORD.LT. CSTBRK) CALL GUM(ISHORD,IBQ,N)
129     GO TO 200
130     100     IDVORD=IFIX(DVORD)
131     IF(DVORD.GE. CSTBRK) GO TO 150
132     IBOPSM(1)=IBOPSM(1) + 1
133     IBOPSM(2)=IBOPSM(2) + IBQ
134     IBOPSM(3)=IBOPSM(3) + IDVORD

```

```
05      GO TO 200
06      150      IBOPLG(1)=IBOPLG(1) +1
07      IBOPLG(2)=IBOPLG(2) + IBO
08      IBOPLG(3)=IBOPLG(3) + IDVORD
09      200      CONTINUE
10      IF(IDBUG.NE.1) GO TO 22
11      BUYDOL = UCOST(N)*IBO
12      WRITE(6,8000)N,IBO,INVDUE(N),NLGC,UCOST(N),BUYDOL
13      8000 FORMAT("*****ORDER=#N#",I3," IBO=",I5," INVDUE=",I5,
14      8" NLGC=",I5," UCOST=",F10,2,T110," BUY$="
15      A,F12,2)
16      22 CONTINUE
17      RETURN
18      END
```

Subroutine: OUTREP

Function:

This routine presents a short-form summary report of simulation results obtained from a given RIME simulation run.

Description:

This routine produces a short-form summary report of RIME simulation results. The quarterly simulation statistics are totalled in this routine for selected measures, and printed in a short-form (3 pages) report, presenting a compact summary of major performance statistics of interest in this study. See Volume I, for a detailed discussion of the output report produced by this routine.

```

1      *NRUN=,RIME/OBJ/OUTREP,0(IGD,NOG6)
2      *OUTREP,S RIME SHORT-FORM SUMMARY REPORT
3      SUBROUTINE OUTREP(KF,KL)
4          COMMON/IDENT/IDENT
5          COMMON/MLAM/MLAM,MLAM
6          COMMON/MREPL/MREPL,MREPL
7          COMMON/NGROUP/NGROUP
8          COMMON/MFGRP/MFGRP
9          COMMON/IORDER/ IORDER(16,3,6)
0          COMMON/IWFP/ IWFP(16,3,6)
1          COMMON/ITBKDT/ITBKDT(16,3,6)
2          COMMON/IBODAI/IBODAI(16,3,6)
3          COMMON/IBODAT/IBODAT(16,3,6)
4          COMMON/IREOT/IREOT(16,3,6)
5          COMMON/IFILLT/IFILLT(16,3,6)
6          COMMON/INQTR/ INQTR
7          DIMENSION FIL(6)
8          DIMENSION ITORD(6),ITWFP(6),ITBKDT(6),ITREQ(6),ITFILL(6)
9          DIMENSION TYPE(3,3)
10         DATA TYPE/"ACTI","ONS/","FSN ","UNIT","S ","","DOLL","ARS ","
11         " "/"
12         &
13         C          PRODUCE REPORT FOR MEASURES J=1,3
14         C          WHERE J=1  ACTIONS/FSN
15         C          J=2  UNITS
16         C          J=3  DOLLARS
17         C
18         DO 1000 J=KF,ML
19         C
20         C          PRINT HEADINGS:
21         C
22         WRITE(6,60)(TYPE(K,J),K=1,3),J
23         60 FORMAT("1"///T20,"SIMULATION RESULTS IN ".5A4.2X,"(J=",11," )")
24         WRITE(6,63)NGROUP,IDENT,MLAM,MREPL,INQTR
25         63 FORMAT(/T10,"NGROUP =",I3," IDENT =",I3,
26         " MLAM =",I3," MREPL =",I3," INQTR =",I3,
27         &
28         &
29         C
30         C          INITIALIZE ARRAYS AND VARIABLES,
31         C
32         DO 20 K=1,6
33         ITORD(K)=0
34         ITWFP(K)=0
35         ITBKDT(K)=0
36         ITREQ(K)=0
37         ITFILL(K)=0
38         20 CONTINUE
39         IGDORD=0
40         IGDWFP=0
41         IGDBAK=0
42         IGDREQ=0

```

OUTREP

```

53      IGBFIL=0
54      C
55      C      COMPUTE SUMMATIONS FOR INOTR QUARTERS FOR EACH OF THE
56      C      AGGREGATE CATEGORIES(K=1,6).
57      C      WHERE K=1 REPRESENTS LRU'S AT BASE
58      C      K=2 REPRESENTS SRU'S AT BASE
59      C      K=3 REPRESENTS LRU'S AT DEPOT
60      C      K=4 REPRESENTS SRU'S AT DEPOT
61      C      K=5 REPRESENTS LRU'S AT OVERHAUL FACILITY
62      C      K=6 REPRESENTS SRU'S AT OVERHAUL FACILITY
63      C
64      DO 100 K=1,6
65      DO 50 I=1,INOTR
66      ITORD(K)=ITORD(K)+IORDER(I,J,K)
67      ITWFP(K)=ITWFP(K)+IWFP(I,J,K)
68      ITBAKD(K)=ITBAKD(K)+ITBAKDT(I,J,K)
69      ITREQ(K)=ITREQ(K)+IREQT(I,J,K)
70      ITFILL(K)=ITFILL(K)+ITFILLT(I,J,K)
71      50 CONTINUE
72      C
73      C      COMPUTE GRAND TOTALS.
74      IGDORD=IGDORD+ITORD(K)
75      IGDWFP=IGDWFP+ITWFP(K)
76      IGDBAK=IGDBAK+ITBAKD(K)
77      IGDREQ=IGDREQ+ITREQ(K)
78      IGBFIL=IGBFIL+ITFILL(K)
79      100 CONTINUE
80      C
81      C      COMPUTE INTERMEDIATE TOTALS FOR "ORDERS PLACED."
82      C
83      INTOD1=ITORD(1)+ITORD(2)
84      INTOD2=ITORD(3)+ITORD(4)
85      INTOD3=ITORD(5)+ITORD(6)
86      C
87      C      COMPLETE INTERMEDIATE TOTALS FOR "WAIT FOR PARTS".
88      C
89      INTWP1=ITWFP(1)+ITWFP(2)
90      INTWP2=ITWFP(3)+ITWFP(4)
91      INTWP3=ITWFP(5)+ITWFP(6)
92      C
93      C      COMPUTE INTERMEDIATE TOTALS FOR "BACKORDER WEEKS".
94      C
95      INTBK1=ITBAKD(1)+ITBAKD(2)
96      INTBK2=ITBAKD(3)+ITBAKD(4)
97      INTBK3=ITBAKD(5)+ITBAKD(6)
98      C
99      C      COMPUTE INTERMEDIATE TOTALS FOR "TOTAL REQUISITIONS".
100     C
101     INTRO1=ITREQ(1)+ITREQ(2)
102     INTRO2=ITREQ(3)+ITREQ(4)
103     INTRO3=ITREQ(5)+ITREQ(6)
104     C

```

```

5      C      COMPUTE INTERMEDIATE TOTALS FOR "TOTAL FILLS".
6      C
7      INTFL1=ITFILL(1)+ITFILL(2)
8      INTFL2=ITFILL(3)+ITFILL(4)
9      INTFL3=ITFILL(5)+ITFILL(6)
0
1      C
2      C      COMPUTE "FILL PERCENTAGES".
3
4      C
5      DO 200 K=1,6
6      FIL(K)=FILLF(ITFILL(K),ITREQ(K))
7      200 CONTINUE
8      FILP1=FILLF(INTFL1,INTREQ1)
9      FILP2=FILLF(INTFL2,INTREQ2)
10     FILP3=FILLF(INTFL3,INTREQ3)
11     FILGD=FILLF(IGDFIL,IGDREQ)
12
13     C
14     C
15     C      PRINT INTERMEDIATE HEADINGS
16     C
17     WRITE (6,210) (I,I=1,6)
18     210 FORMAT(///,17X,6("I",I1,"",7X)/
19     &      /,11X,"REQUISITIONS",4X,"WAIT"/
20     &      " ",16X,"TO      FOR BACKORDER  TOTAL      TOTAL  ",
21     &      " FILL"/,
22     &      " ",13X,"SUPPLIERS  PARTS  WEEKS  REQUISITIONS FILLS",
23     &      " PERCENTAGE"/
24     &      " ",13X,6("-----",2X))
25
26     C
27     C      WRITE OUT RESULTS.
28     C
29     WRITE(6,250) ITORD(1),ITWFP(1),ITBAKD(1),ITREQ(1),
30     &      ITFILL(1),FIL(1)
31     250 FORMAT(///,"BASE LRU "5X10,F10.3)
32
33     WRITE(6,260) ITORD(2),ITWFP(2),ITBAKD(2),ITREQ(2),ITFILL(2),FIL(2)
34     260 FORMAT(///,"BASE SRU "5X10,F10.3)
35     WRITE(6,270) INTOD1,INTWP1,INTBK1,INTREQ1,INTFL1,FILP1
36     270 FORMAT(" ",13X,6("-----",2X)/
37     &      /" TOTAL",5X10,F10.3)
38     WRITE(6,280) ITORD(3),ITWFP(3),ITBAKD(3),ITREQ(3),ITFILL(3),FIL(3)
39     280 FORMAT(///,"DEPOT LRU",5X10,F10.3)
40     WRITE(6,290) ITORD(4),ITWFP(4),ITBAKD(4),ITREQ(4),ITFILL(4),FIL(4)
41     290 FORMAT(///,"DEPOT SRU",5X10,F10.3)
42     WRITE(6,270) INTOD2,INTWP2,INTBK2,INTREQ2,INTFL2,FILP2
43     WRITE(6,330) ITORD(5),ITWFP(5),ITBAKD(5),ITREQ(5),ITFILL(5),FIL(5)
44     330 FORMAT(///,"OVERHL LRU",5X10,F10.3)
45     WRITE(6,340) ITORD(6),ITWFP(6),ITBAKD(6),ITREQ(6),ITFILL(6),FIL(6)
46     340 FORMAT(///,"OVERHL SRU",5X10,F10.3)
47     WRITE(6,270) INTOD3,INTWP3,INTBK3,INTREQ3,INTFL3,FILP3
48     WRITE(6,320) IGDORD,IGDWFP,IGDBAK,IGDREQ,IGBFIL,FILGD
49     320 FORMAT(///,"GRAND TOT", 5X10,F10.3/
50     &      " ",13X,6("-----",2X)/

```

02 01 10-20-79 10,275 .5 RIME SHORT-FORM SUMMARY REPORT

```

157      &          " ",13X,6("-----",2X))
158      1000 CONTINUE
159      C
160      C          PRINT BACKORDER=DAY STATISTICS
161      C
162      WRITE(6,1113)
163      1113 FORMAT(1H1,///T30,"BACKORDER=DAY STATISTICS",//)
164      WRITE(6,1114)
165      1114 FORMAT(T17,"BASE",T37,"DEPOT",T57,"OVER-HAUL",
166      &          T77,"BASE",T97,"DEPOT",T117,"OVER-HAUL",
167      &          /T16X3("IBODAT",14X),2("IBODAT",14X),"IBODAT",/
168      &          T6,6("      LRU      SRU")/
169      &          )
170      DO 1140 J=1,3
171      WRITE(6,1123)J
172      1123 FORMAT(T10,"J =",I3)
173      DO 1130 I=1,INQTR
174      WRITE(6,1133)I,(IBODAT(I,J,K),K=1,6),(IBODAT(I,J,K),
175      &          K=1,6)
176      1133 FORMAT(I5,12I40)
177      1130 CONTINUE
178      1140 CONTINUE
179      C
180      RETURN
181      END

```

01 10-20-79 10.276 -COMPUTE FILL FRACTION

```
1 *FILLF--COMPUTE FILL FRACTION
2 FUNCTION FILLF(ITOP,IBOT)
3   DENOM=FLOAT(IBOT)
4   IF (DENOM.LT.1.)DENOM=1.
5   FILLF=FLOAT(ITOP)/DENOM
6   RETURN
7   END
```

FILLF

Subroutine: OUT2

Function:

This routine outputs summary statistics by quarter to summarize simulation results. Values printed represent the sum of all statistics collected for all LRU/SRU groups and all associated replications performed in the current RIME run.

Description:

This routine outputs summary statistics describing the detailed results of a RIME simulation run. See Volume I for a detailed description for the outputs of this routine.

1T 01 10-20-79 1.551

```

1      *#RUN=IRIME/BBB/OUT2.0(BCD,NOGO)
2      *OUT2.0
3      SUBROUTINE OUT2(KF,KL)
4      C      THIS ROUTINE OUT2.0 SUMMARY STATISTICS FOR STATISTICS
5      C      CATEGORIES KF THRU KL, WHERE
6      C      1=LRU'S AT BKSB
7      C      2=SRU'S AT BKSB
8      C      ETC.
9      DIMENSION ITOTL(29 ),AVEYR(29 )
10     DIMENSION TEXT(5,6)
11     C
12     COMMON/ELAM/ELAM,MLAM
13     COMMON/NREPL/NREPL,NREPL
14     COMMON/NGROUP/NGROUP
15     C
16     COMMON/KNQTR/KNQTR
17     COMMON/ITIME/ITIME
18     COMMON/ITYEAR/ITYEAR
19     COMMON/ITINV/ITINV
20     COMMON/INVACT/INVACT(1)
21     COMMON/NDENT/NDENT(1)
22     C
23     COMMON/NITEM/NITEM
24     COMMON/IBOP/IBOPON(3),IBOPON(3)
25     COMMON/IBACKI/IBACKI(16,3,6)
26     COMMON/IBACKT/IBACKT(16,3,6)
27     COMMON/IBAKDI/IBAKDI(16,3,6)
28     COMMON/IBAKDT/IBAKDT(16,3,6)
29     COMMON/ICANCL/ICANCL(16,3,6)
30     COMMON/IDISPS/IDISPS(16,3,6)
31     COMMON/IXPED/IXPED(16,3,6)
32     COMMON/IFILLI/IFILLI(16,3,6)
33     COMMON/IFILLT/IFILLT(16,3,6)
34     COMMON/IORDER/IORDER(16,3,6)
35     COMMON/INVDAY/INVDAY(16,3,6)
36     COMMON/INVOH/INVOH(16,3,6)
37     COMMON/INVOB/INVOB(16,3,6)
38     COMMON/IRATON/IRATON(16,3,6)
39     COMMON/IRECET/IRECET(16,3,6)
40     COMMON/IREOC /IREOC(16,3,6)
41     COMMON/IREOI /IREOI(16,3,6)
42     COMMON/IREOT/IREOT(16,3,6)
43     COMMON/IRETRN/IRETRN(16,3,6)
44     COMMON/ISHIPT/ISHIPT(16,3,6)
45     COMMON/ISHIPT/ISHIPT(16,3,6)
46     COMMON/ITERM /ITERM(16,3,6)
47     C-----REPARABLE COMMONS
48     COMMON/IREPON/IREPON(16,3,6)
49     COMMON/ICNDEM/ICNDEM(16,3,6)
50     COMMON/INRTS/INRTS(16,3,6)
51     COMMON/IRECPL/IRECPL(16,3,6)
52     COMMON/IWIP/IWIP(16,3,6)

```

OUT2

```

53      COMMON/INFP/INFP(16,3,8)
54      C-----
55      C
56      C
57      DATA TEXT/"LEU ", "AT B", "ASE ", "FACE", "LITY",
58      &          "SRU ", "AT B", "ASE ", "FACE", "LITY",
59      &          " ", "LEU ", "AT B", "EPOT", " ",
60      &          " ", "SRU ", "AT B", "EPOT", " ",
61      &          " LR ", "U IN", " SVS", "RNAU", "L ",
62      &          " SR ", "U IN", " SVS", "RNAU", "L  "/
63      C
64      C      LOOP THROUGH LOCATION CATEGORIES K
65      C
66      R=FLOAT(ITIME)/FLOAT(IYEAR)
67      C
68      C      PRINT REPORT HEADER
69      C
70      WRITE(6,100)
71      WRITE(6,102)
72      WRITE(6,103)
73      WRITE(6,102)
74      1 CONTINUE
75      DO 560 K=KF,KE
76      C
77      C
78      40 WRITE(6,122)(TEXT(KK,K),KK=1,5)
79      WRITE(6,123)NGROUP,MREPL,MLAM
80      123 FORMAT(10C,"NGROUP =",I3," MREPL =",I3,
81      &          " MLAM =",I3)
82      C
83      C
84      C      WRITE OUT COLUMN HEADINGS
85      C
86      WRITE(6,108)
87      DO 260 J=1,2
88      JJ=J
89      CALL STATED(JJ,ITOTL)
90      DO 50 I=1,INOTS
91      ITOTL(1)=ITOTL(1)+INVON(I,J,K)
92      ITOTL(2)=ITOTL(2)+INVR(I,J,K)
93      ITOTL(3)=ITOTL(3)+IRECET(I,J,K)
94      ITOTL(4)=ITOTL(4)+IRETRN(I,J,K)
95      ITOTL(5)=ITOTL(5)+ISHIPT(I,J,K)
96      ITOTL(6)=ITOTL(6)+ISHIPT(I,J,K)
97      ITOTL(7)=ITOTL(7)+IORDER(I,J,K)
98      ITOTL(8)=ITOTL(8)+IREQT(I,J,K)
99      ITOTL(9)=ITOTL(9)+IREQC(I,J,K)
100     ITOTL(10)=ITOTL(10)+IREQI(I,J,K)
101     50 WRITE(6,120)I,INVON(I,J,K),INVR(I,J,K),IRECET(I,J,K),
102     &      IRETRN(I,J,K),
103     &      ISHIPT(I,J,K),ISHIPT(I,J,K),IORDER(I,J,K),IREQT(I,J,K),
104     &      IREQC(I,J,K),IREQI(I,J,K)

```

31T 01 10-20-79 10.551

```

105      WRITE(6,110)
106      WRITE(6,202) (ITOTL(I),I=1,40)
107      202 FORMAT(1H,8H*TOTALS*,T20,10X50)
108      C
109      C      COMPUTE AVERAGES
110      C
111      DO 203 I=1,40
112      203 AVEYR(I)=FLOAT(ITOTL(I))/R
113      WRITE(6,204) (AVEYR(I),I=1,40)
114      204 FORMAT(1H,6HAYE/YR,T21,10X10.0)
115      260 CONTINUE
116      WRITE(6,122) (TEXT(KK,K),KK=1,5)
117      WRITE(6,123) NGROUP,MREPL,MLXN
118      WRITE(6,111)
119      DO 360 J=1,3
120      JJ=J
121      CALL STATND(JJ,ITOTL)
122      C
123      C      ACCUMULATE OVER ALL QUARRERS
124      C
125      DO 57 I=1,INOTE
126      ITOTL(11)=ITOTL(11)+IEXPED(I,J,K)
127      ITOTL(12)=ITOTL(12)+INATON(I,J,K)
128      ITOTL(13)=ITOTL(13)+IDISPS(I,J,K)
129      ITOTL(14)=ITOTL(14)+ITERM(I,J,K)
130      ITOTL(20)=ITOTL(20)+IREPGN(I,J,K)
131      ITOTL(25)=ITOTL(25)+ICNDEM(I,J,K)
132      ITOTL(26)=ITOTL(26)+INRSTS(I,J,K)
133      ITOTL(27)=ITOTL(27)+INRCP(I,J,K)
134      ITOTL(28)=ITOTL(28)+IWIP(I,J,K)
135      ITOTL(29)=ITOTL(29)+IWFP(I,J,K)
136      57 WRITE(6,120) I,IEXPED(I,J,K),INATON(I,J,K),IDISPS(I,J,K),
137      & ITERM(I,J,K),IREPGN(I,J,K),ICNDEM(I,J,K),INRSTS(I,J,K),INRCP(I,J,K),
138      & IWIP(I,J,K),IWFP(I,J,K)
139      WRITE(6,67)
140      WRITE(6,205) (ITOTL(I),I=11,18) & (ITOTL(I),I=24,29)
141      205 FORMAT(T12,'*TOTAL*',T20,10X10)
142      C
143      C      COMPUTE AVERAGES
144      C
145      DO 206 I=11,18
146      206 AVEYR(I)=FLOAT(ITOTL(I))/R
147      DO 220 I=24,29
148      220 AVEYR(I)=FLOAT(ITOTL(I))/R
149      WRITE(6,207) (AVEYR(I),I=11,18),(AVEYR(I),I=24,29)
150      207 FORMAT(T13,'AVE/YR',T21,10X10.0)
151      360 CONTINUE
152      C
153      C      WRITE OUT COLUMN HEADINGS
154      C
155      WRITE(6,122) (TEXT(KK,K),KK=1,5)
156      WRITE(6,123) NGROUP,MREPL,MLXN

```

1T 01 10-20-79 10.554

```

157      WRITE(6,115)
158      WRITE(6,68)
159      DO 460 J=1,3
160          JJ=J
161          CALL STATED(JJ,ITOTL)
162      C      DETERMINE NUMBER OF REQUESTIONS FROM CUSTOMERS
163      C      EACH QUARTER
164          DO 58 I=1,INOTE
165              IF(IREQ(I,J,K).LE.0) GO TO 62
166              T1=FLOAT(IPILLT(I,J,K))/FLOAT(IREQ(I,J,K))
167              GO TO 63
168          62 T1=0.
169          63 CONTINUE
170              IF(IREQ(I,J,K).LE.0) GO TO 64
171              T2=FLOAT(IPILLI(I,J,K))/FLOAT(IREQ(I,J,K))
172              GO TO 65
173          64 CONTINUE
174              T2=0.
175          65 CONTINUE
176              ITOTL( 8)=ITOTL(8)+IREQ(I,0,K)
177              ITOTL(10)=ITOTL(10)+IREQ(I,J,K)
178              ITOTL(15)=ITOTL(15)+IBACKT(Z,0,K)
179              ITOTL(16)=ITOTL(16)+IBACKY(Z,0,K)
180              ITOTL(17)=ITOTL(17)+IBAKDT(Z,0,K)
181              ITOTL(18)=ITOTL(18)+IBAKDY(Z,0,K)
182              ITOTL(19)=ITOTL(19)+INVDAY(Z,0,K)
183              ITOTL(20)=ITOTL(20)+IPILLY(Z,0,K)
184              ITOTL(21)=ITOTL(21)+IPILLI(Z,0,K)
185          58 WRITE(6,117) I,IBACKT(I,J,K),IBACKY(I,J,K),IBAKDT(I,J,K),
186      &      IBAKDY(I,J,K),INVDAY(I,J,K),IPILLT(I,J,K),IPILLI(I,J,K),T1,T2
187              WRITE(6,68)
188              WRITE(6,208) (ITOTL(I),I=15,25)
189          208 FORMAT(1H0,8X,8H"TOTALS",1X,7F10.2X)
190              DO 209 I=15,21
191          209 AVEYR(I)=FLOAT(ITOTL(I))/N
192              AVEYR(22)=0.
193              AVEYR(23)=0.
194              IF(ITOTL(8).GT.0) AVEYR(22)=FLOAT(ITOTL(20))/FLOAT(ITOTL(8))
195              IF(ITOTL(10).GT.0) AVEYR(23)=FLOAT(ITOTL(21))/FLOAT(ITOTL(10))
196          210 FORMAT(8X,7H"AVE/YR",7F12.6,7F16.3)
197              WRITE(6,210) (AVEYR(I),I=15,23)
198          460 CONTINUE
199      C
200      C      INCREMENT AGGREGATE CATEGORY
201      C
202          560 CONTINUE
203          900 RETURN
204              67 FORMAT( T12,11(8(1H-9.2X))
205              68 FORMAT(1H :1X,2H=-,2X,7(12H -----,2(14H -----))
206          100 FORMAT(1H1./2./3./)
207          101 FORMAT(1H./2./)
208          102 FORMAT(1H.48X,44(1H-))

```

31T 01 10-20-79 1.554

```

209      103 FORMAT(1H ,48X,44HP E R F O R M A N C E   S T A T I S T I C S)
210      110 FORMAT(1H ,22,10(8(1H=),2X))
211      *
212      108 FORMAT(//,
213      & T22,'INVENTORY INVENTORY',T64,'TOTAL      PRT I',
214      & T83,'ORDERS      TOTAL      REQD      PRIORITY I',
215      & T13,'PERIOD      ON-HAND      ON-ORDER RECEIPTS',
216      & T53,'RETURNS  SHIPMENTS SHIPMENTS  PLACED',
217      & T95,'REQS  CANCELED  REQSD',
218      & T12,11(8(1H=),2X))
219      111 FORMAT(//,
220      & T32,'RATIONING',
221      & T92,'COMPLETED REQ WORK WAIT FOR',
222      & T13,'PERIOD      EXPEDITES  ACTIONS',
223      & T42,'DISPOSALS  TERMIN  REP GNS',
224      & T72,'CONDEMNNS  NRTS      REPAIRS IN',
225      & T14,'PROCESS  PARTS',
226      & T12,11(8(1H=)2X))
227      115 FORMAT(//,
228      & T21,'TOTAL      PRIORITY I      TOTAL ',
229      & T55,'      PRT I      INVENTORY      TOTAL',
230      & T93,'PRIORITY I  TOT FILLS  PRT I FILLS',
231      & T17,'PERIOD  BACKORDERS  BACKORDERS',
232      & T45,'  90-DAYS      90-DAYS      90ERS',
233      & T84,'FILLS      FILLS      /TOT REQ      /PRT I REQSD',
234      117 FORMAT(1H ,T43,I2,T17,7(2X,I18),3X,2(1X,I10,2))
235      120 FORMAT(1H ,T43,I2,T20,10(2X,I6))
236      121 FORMAT(1H ,1X,3NDOP,2X,2(4X,I6))
237      122 FORMAT(1H1,56X,5A4/1H ,56X,10Z1H*)
238      END
**W      7 MEMORY EXPANDED. USE $LIMITS OR CONE= OPTION FOR NEXT RUN

```

Subroutine: RCVPRTFunction:

Event Code 17. This routine records the receipt of IQTY units of SKU N to support the repair of LRU reparable generation number NJOB. If all parts requirements for the repair of the LRU are satisfied, the routine schedules a repair completion event (Event Type 18) for this LRU.

Calling Parameters:

N = The SKU of inventory item associated with this receipt

IQTY = The number of assets being received

NJOB = The job number associated with the LRU to be repaired

Description:

Subroutine RCVPRT first utilizes the GASP routine NFIND to locate the work-in-process record for the LRU reparable generation number NJOB. The total number of parts needed to complete the repair of the LRU (stored in the GASP vector QSET(NAT2)) is then reduced by the number of assets IQTY that has just been received. If all parts needs have now been satisfied (i.e., if QSET(NAT2) is

now zero), the work-in-process record is permanently removed from GASP file 2 by calling GASP routine RMOVE.

When all parts needs for completion of the LRU repair have been satisfied, subroutine CUM is called to update the waiting time statistics array IWFP. In addition, RCVPRT then schedules a repair completion event (Event Type 18) for the LRU.

```

1      SUBROUTINE RCVPRT(N,IQTY,NJOB)
2      C      EVENT CODE 17. RECORD RECEIPT OF IQTY UNITS OF ITEM N
3      C      TO SUPPORT REPAIR FOR NJOB. IF ALL PARTS NEEDS ARE
4      C      NOW SATISFIED, SCHEDULE BEGINNING OF REPAIR (EVENT
5      C      CODE 18) FOR THIS ITEM.
6      C
7      COMMON /GCOM1/ ATTRIB(30),JEVNT,MFA,MFE(100),MLE(100),MSTOP,MCRDR,
8      &  MWABC,MWAPT,MWATR,MWFIL,MWO(100),MNTY,MPRINT,PPARM(50,4),TNOW,TTSEG,
9      &  TTCLR,TTFIN,TTIRB(30),TTSET
10     COMMON /GCOM6/ EENO(100),XINW(100),KKRNM(100),MMAHQ(100),
11     &  QOTIM(100),SSOBY(25,5),SSTPV(25,6),VVNC(100)
12     DIMENSION NSET(1)
13     COMMON QSET(1)
14     EQUIVALENCE (NSET(1),QSET(1))
15     C
16     COMMON/IDBUG/IDBUG
17     COMMON/ITDAY/ITDAY
18     COMMON/ITIME/ITIME
19     COMMON/ITWEEK/ITWEEK
20     COMMON/IWFP/IWFP(16,3,6)
21     COMMON/IDORT/IDORT(1)
22     COMMON/IDRT/IDRT(1)
23     COMMON/IBRT/IBRT(1)
24     C
25     IF(IDBUG.NE.1) GO TO 20
26     WRITE(6,33)ITIME,N,IQTY,NJOB
27     20 CONTINUE
28     C
29     C      LOCATE ENTRY NUMBER NTRY OF JOB NJOB IN THE BACKORDER FI
30     C      (FILE 2). JOB NUMBERS ARE RECORDED IN ATTRIBUTE 3
31     C
32     XVAL=NJOB
33     NTRY=NFIND(XVAL,5,2,3,.2)
34     IF(NTRY.GT.0)GO TO 100
35     C
36     C      IF NTRY IS ZERO, NO ENTRY WAS FOUND.
37     C      PRINT AN ERROR MESSAGE, AT RETURN.
38     C
39     WRITE(6,33) ITIME,N,IQTY,NJOB
40     33 FORMAT('*****RCVPRT      ITIME=','I8,' N=','I8,' IQTY=','I8,
41     &  ' NJOB=','I8)
42     WRITE(6,13)'-----NO PARTS RECORD FOR NJOB WAS FOUND'
43     13 FORMAT(V)
44     RETURN
45     C
46     C      DECREASE PARTS NEEDS TO REFLECT THIS RECEIPT.
47     C      NEEDS ARE STORED IN ATTRIBUTE 2.
48     C
49     100C CONTINUE
50     NAT2=NTRY+2
51
52     QSET(NAT2)=QSET(NAT2)-IQTY

```

RCVPRT

T 01 10-20-79 10,301

```

53      IF(IDBUG.EQ.1)WRITE(6,13) "---PARTS STILL NEEDED=",QSET(NAT2)
54      C
55      C      IF PARTS ARE STILL NEEDED: RETURN
56      C
57      IF(QSET(NAT2).GT.0.)RETURN
58      C
59      C      ALL PARTS NEEDS HAVE BEEN SATISFIELD:
60      C      REMOVE THIS JOB FROM WAIT-FOR-PARTS FILE.
61      C
62      CALL XMOVE(NTBY,2)
63      N=ATRI(1)
64      NJOB=ATRI(3)
65      IWAIT=(ITIME-ATRI(4))
66      C
67      C      RECORD TIME WAITING FOR PARTS
68      C
69      CALL CUM(IWFR,IWAIT,N)
70      C
71      C      DETERMINE IF THIS REPAIR SITE IS A DEPOT(1 OR 2),A
72      C      BASE (3 OR 4), OR OVERHAUL LOCATION(5 OR 6)
73      C
74      JTIME=0
75      NCODE=KNSKU(N)
76      C
77      C      IF DEPOT
78      C
79      IF ((NCODE.EQ.5).OR.(NCODE.EQ.6))JTIME=IDBT(N)*ITDAY
80      C
81      C      IF A BASE LOCATION
82      C
83      IF((NCODE.EQ.3).OR.(NCODE.EQ.4))JTIME=IBRT(N)*ITDAY
84      C
85      C      IF AN OVERHAUL FACILITY
86      C
87      IF((NCODE.EQ.1).OR.(NCODE.EQ.2))JTIME=IDBT(N)*ITDAY
88      C
89      C      SCHEDULE REPAIR COMPLETION
90      C
91      JTIME=JTIME + ITIME
92      CALL ENTER(JTIME,18,N,IOTY,NJOB)
93      RETURN
94      END

```

Subroutine: RECEIV

Function:

Event Type 2. This routine updates stock status records to reflect the receipt of a replenishment order from a supplier of the inventory system.

Calling Parameters:

N = The SKU of the inventory location receiving the replenishment order.

IQTY = The number of assets received.

Description:

The routine first calls subroutine CUM to update the statistics array IRECET. The on hand and due in inventory arrays for SKU N are then updated. Finally, subroutine FILLBO is called to fill outstanding backorders for SKU N, if any.

```
1  *#RUN:=TIME/OBJ/RECEIV.0(BCD,NOG6)
2  *RECEIV,S      EVT. CODE 2.  RECEIVE EVENT
3  SUBROUTINE RECEIV(N,IQTY)
4  C      THIS ROUTINE UPDATES STOCK STATUS RECORDS TO REFLECT RECEIPT
5  C      OF AN ORDER FOR IQTY UNITS OF ITEM N DURING PERIOD I.
6  C      THE ROUTINE THEN INITIATES SHIPMENT ACTIONS TO FULFILL
7  C      OUTSTANDING BACKORDERS      IF ANY.
8  COMMON/IDBUG/IDBUG
9  COMMON/NLOCBK/NLOCBK
10 COMMON/IRECET/IRECET(16,3,6)
11 COMMON/ISHIPT/ISHIPT(16,3,6)
12 COMMON/ISHIPT/ISHIPT(16,3,6)
13 COMMON/JPRIOR/JPRIOR(500)
14 COMMON/IQTYB/IQTYB(200)
15 COMMON/IPRIOR/IPRIOR(200)
16 COMMON/NBOTU/NBOTU(4)
17 COMMON/ILOCBK/ILOCBK(200)
18 COMMON/INVACT/INVACT(1)
19 COMMON/INVDUE/INVDUE(1)
20 COMMON/NBOPT/NBOPT(1)
21 COMMON/ISUL/ISUL(1)
22 COMMON/IBACPT/IBACPT(200)
23 C      ADJUST GROSS RECEIPT STATISTICS
24 CALL CUM(IRECET,IQTY,N)
25 C      ADJUST      ASSETS FOR THIS      ITEM
26 INVACT(N)=INVACT(N)+IQTY
27 INVDUE(N)=INVDUE(N)-IQTY
28 C      ARE THERE ANY BACKORDERS ON THIS ITEM.
29 IF(NBOPT(N).LE.0) RETURN
30 CALL FILBO(N)
31 RETURN
32 END
```

RECEIV

Subroutine: REMOVE

Function:

This routine removes the earliest event transaction from the Future Events List, and updates associated pointer records.

Description:

This routine determines the "current" event to be simulated within RIME. For a detailed description of the event bookkeeping process, see Volume I, Section II.

```
1      SUBROUTINE REMOVE(KTIME,KTYPE,KFSN,KQTY,KPRIOR)
2      C      THIS ROUTINE REMOVES THE EARLIEST TRANSACTION FROM THE
3      C      CURRENT EVENTS CHAIN, AND UPDATES THE CHAIN STRUCTURE
4      COMMON/IDBUG/IDBUG
5      COMMON/NENTRY/NENTRY
6      COMMON/NFIRST/NFIRST
7      COMMON/NTIME/NTIME
8      COMMON/ILOCFE/ILOCFE(1)
9      COMMON/JFSN/JFSN(1)
10     COMMON/JPOINT/JPOINT(1)
11     COMMON/JPRIOR/JPRIOR(1)
12     COMMON/JQTY/JQTY(1)
13     COMMON/JTIME/JTIME(1)
14     COMMON/JTYPE/JTYPE(1)
15     K=NFIRST
16     KTIME=JTIME(K)
17     KTYPE=JTYPE(K)
18     KFSN=JFSN(K)
19     KQTY=JQTY(K)
20     KPRIOR=JPRIOR(K)
21     NFIRST=JPOINT(K)
22     NTIME=JTIME(NFIRST)
23     IF(IDBUG.NE.1) GO TO 25
24     WRITE(6,8000)KTIME,KTYPE,KFSN,KQTY,KPRIOR,K
25     8000 FORMAT(7H REMOVE,I10,5I8)
26     25 CONTINUE
27     C      IS THIS THE LAST TRANSACTION ON THE LIST
28     IF(NENTRY.GT.1) GO TO 19
29     C      SET LIST CLOCK UP A LARGE INCREMENT
30     NTIME=99999999
31     C      RECORD LOCATION K AS AVAILABLE
32     19 ILOCFE(NENTRY)=K
33     C      DECREMENT COUNT OF LIST ENTRIES
34     NENTRY=NENTRY-1
35     RETURN
36     END
```

REMOVE

Subroutine: REPGEN

Function:

Event Type 14. This routine records the occurrence of reparable generations, and updates appropriate statistics.

Calling Parameters:

N = The SKU associated with the reparable generation.

IQTY = The number of units associated with this reparable generation.

NJOB = The job number assigned to this reparable generation.

Description:

The routine first calls subroutine CUM to update the reparable generation statistics array IREPGN. Work-in-process for SKU N (INWIP(N)) is then increased by IQTY units, and logic returns to the calling program.

*#RUN=:TIME/OBJ/REPAIR.O(BGD,NOGO)

*REPAIR,S

```
1      SUBROUTINE REPGEN(N,IQTY,NJOB)
2      C      EVENT CODE 14. THIS ROUTINE REGARDS THE GENATION OF
3      C      IQTY UNITS OF SKU N. NJOB IS THE NUMBER ASSIGNED TO
4      C      THIS REP GEN.
5
6      COMMON/IREPGN/IREPGN(16,3,6)
7      COMMON/INWIP/INWIP(1)
8
9      C      UPDATE REP GEN'S STATISTICS
10     C
11     C      CALL CUM(IREPGN,IQTY,N)
12
13     C      UPDATE WORK-IN-PROCESS
14     C
15     INWIP(N)=INWIP(N)+ IQTY
16     RETURN
17     END
```

AD-A121 987

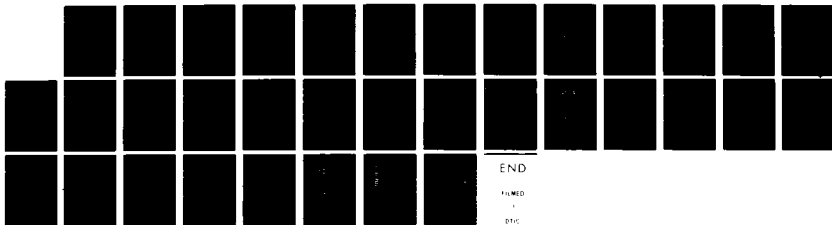
RIME: THE RECOVERABLE ITEM MANAGEMENT EVALUATOR VOLUME
II SECTION III RIM. (U) DECISION SYSTEMS DAYTON OH
W 5 DEMV MAY 80 TR-80-02-C F33600-78-C-0524

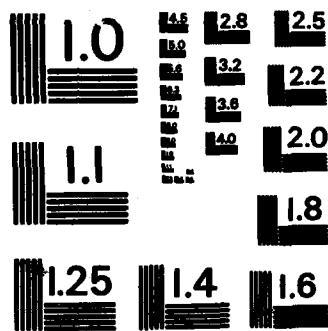
2/2

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Subroutine: REQ

Function:

Event Type 1. This routine reflects the receipt of a customer requisition, and initiates associated shipping and/or backorder actions.

Calling Parameters:

N = The SKU associated with the current requisition event.

IQTY = The number of assets associated with this requisition.

LPRI = A packed variable defining the priority of the requisition and the original source of the requisition.

LTIME = The time of receipt of the current requisition.

Description:

The packed variable LPRI defines both the priority of the current requisition and the original source of the requisition. The priority IPRI is contained in the last two digits of the variable LPRI. Hence, $IPRI = \text{MOD}(LPRI, 100)$. If LPRI is less than 100, the requisition represents an independent exogenous demand originating

from SKU N. If $100 = LPRI = 100,000$, the requisition represents a replenishment order from another SKU. Finally, if $LPRI = 100,000$, the requisition is to provide parts for the repair of job number $(LPRI - IPRI)/100$. When called, subroutine REQ first computes the Stock Keeping Unit number (NSKU) of the inventory location which initiated the requisition. It then calls subroutine CUM to update the requisition statistics arrays IREQT and IREQI. The demand and requisition counters NDEMAC and NREQAC are then updated; however, these latter arrays are not used in the current implementation of RIME.

The routine then determines the extent to which the current requisition can be filled. For low priority requisitions (i.e., requisitions with a priority code = 2) units are shipped until on hand stock reaches the support level (ISUL (N)) for SKU N. For high priority requisitions shipments are made until on hand stock is reduced to zero. If there is insufficient stock to completely satisfy a requisition, subroutine ENTERB is called to record a backorder for the remaining units.

At the conclusion of the routine, subroutine ENTER is called to schedule appropriate parts receipt events. If the requisition is to replenish another stocking location (i.e., if $NSKU \neq N$ and $NSKU - 1000$), subroutine ENTER is called to schedule receipt of these parts (Event Type 2). On the other hand, if $NSKU = 1000$, the requisition is to provide support for the repair of LRU NSKU. In this case, subroutine ENTER is called to schedule an LRU receive parts event (Event Type 17).

At the conclusion of the routine, subroutine CUM is called to update the shipping and fill rate statistics ISHIPT, ISHIPT, IFILLP, and IFILLI.

```

1  *SUBROUTINE REQ(N,IQTY,LPRI,LTIME)
2  SUBROUTINE REQ(N,IQTY,LPRI,LTIME)
3  C      THIS ROUTINE REFLECTS PROCESSING OF A REQUISITION FOR IQTY
4  C      UNITS OF ITEM N DURING PERIOD I, WHERE IPRI=1 DENOTES A N
5  C      PRIORITY REQUISITION, IPRI=2 DENOTES A LOW PRIORITY REQUISITION.
6  C      JTIME DENOTES THE CLOCK TIME THE REQ WAS RECEIVED
7  C      RECORD REQUISITION DATA.
8  C      IF THIS REQUISITION IS TO PROVIDE PARTS FOR LRU
9  C      REPAIR, LPRI EQUALS (100*NJOB + PRIORITY CODE).
10 C
11 C      RECORD REQUISITION DATA
12 COMMON/ITIME/ITIME
13 COMMON/ITDAY/ITDAY
14 COMMON/LTPROD/LTPROD(1)
15 COMMON/REQST1/REQST1(1)
16 COMMON/REQHAB/REQHAB(1)
17 COMMON/NDEMAC/NDEMAC(1)
18 COMMON/NREQAC/NREQAC(1)
19 COMMON/IREQ1/IREQ1(1)
20 COMMON/IREQ2/IREQ2(1)
21 COMMON/INVACT/INVACT(1)
22 COMMON/ISUL/ISUL(1)
23 COMMON/ISHIPT/ISHIPT(1)
24 COMMON/IFILL1/IFILL1(1)
25 COMMON/IFILL2/IFILL2(1)
26
27 C
28 C      COMPUTE THE PRIORITY CODE AND STOCK KEEPING NUMBER
29 C      FOR THIS REQUISITION.
30 C
31 IPRI=MOD(LPRI,100)
32 NSKU=(LPRI-IPRI)/100
33 IF(NSKU.LE.0) NSKU=N
34 C
35 CALL CUM(IREQ1,IQTY,N)
36 C      IS THIS A PRIORITY 1 REQ
37 IF(IPRI.NE.1) GO TO 20
38 CALL CUM(IREQ2,IQTY,N)
39 C      UPDATE DEMAND ACCUMULATOR AND EXPONENTIALLY SMOOTHED
40 NDEMAC(N)=NDEMAC(N)+IQTY
41 NREQAC(N)=NREQAC(N)+1
42 C      IS THERE ANY STOCK ON HAND
43 IF(INVACT(N).GT.0) GO TO 40
44 C      PUT THIS REQUISITION IN BACKORDER STATUS
45 CALL ENTERB(N,IQTY,ERRY,LTIME)
46 RETURN
47 C      SET ITEST=STOCK REMAINING IF THE ORDER WERE FILLED COMPLETELY
48 40 ITEST=INVACT(N)-IQTY
49 C      IS ITEST ABOVE THE SUPPORT LEVEL
50 IF(ITEST.GE.ISUL(N)) GO TO 50
51 C      IS THIS REQ PRIORITY 1
52 IF(IPRI.NE.1) GO TO 60

```

REQ

```

53      C      CAN THE REQUISITION BE      COMPLETELY SATISFIED FROM STOCK 0
54      IF(INVACT(N).GE.IQTY) GO      TO 80
55      C      SHIP ALL REMAINING ON-HAND STOCK IN PARTIAL FULFILLMENT
56      C      OF THIS REQUISITION
57      IQTYS=INVACT(N)
58      C      COMPUTE QUANTITY TO BE      BACKORDERED AND UPDATE PRI-1 STAT
59      ISK=IQTY-IQTY
60      CALL CUM(ISHIPT,IQTY,N)
61      CALL CUMB(IPILLI,IQTY,N)
62      GO TO 65
63      C      ARE ON-HAND ASSETS ABOVE THE SUPPORT LEVEL
64      60 IF(INVACT(N).LE.ISUL(N))      GO TO 70
65      C      SHIP DOWN TO THE SUPPORT LEVEL
66      IQTYS=INVACT(N)-ISUL(N)
67      C      BACKORDER THE REMAINING QUANTITY
68      ISK=IQTY-IQTY
69      65 IF(ISK.GT.0)CALL ENTERB(N,ISK,LPRI,LTIME)
70      C      UPDATE      STOCK STATUS RECORDS
71      INVACT(N)=INVACT(N)-IQTYS
72      C
73      C      IF NSKU > 1000, SCHEDULE A PARTS RECEIPT EVENT,
74      C      (EVENT CODE 17)
75      C
76      JTIME=ITIME + 10
77      IF(NSKU.GT.1000)CALL ENTER(JTIME,17,N,IQTY,NSKU)
78      C
79      C      IF REQUISITION IS TO REPLENISH A LOWER LEVEL,
80      C      SCHEDULE A RECEIVE EVENT(EVENT CODE 2).
81      C
82      JTIME=ITIME + LTPROD(NSKU)*ITDAY
83      IF((NSKU.NE.N).AND.(NSKU.LT.1000))
84      &      CALL ENTER(JTIME,2,NSKU,IQTY,N)
85      C
86      C      UPDATE      GROSS PERFORMANCE STATISTICS
87      IQTYS=-IQTY
88      CALL CUM(ISHIPT,IQTY,N)
89      CALL CUMB(IPILLT,IQTY,N)
90      RETURN
91      70 CONTINUE
92      C      BACKORDER ENTIRE REQUISITION
93      CALL ENTERB(N,IQTY,LPRI,LTIME)
94      RETURN
95      C      SHIP TO FILL THE ENTIRE REQUISITION
96      80 IQTYS=IQTY
97      C      UPDATE      THE STOCK STATUS RECORDS
98      INVACT(N)=INVACT(N)-IQTYS
99      C
100     C      IF NSKU > 1000, SCHEDULE A PARTS RECEIPT EVENT,
101     C      (EVENT CODE 17)
102     C
103     JTIME=ITIME + 10
104     IF(NSKU.GT.1000)CALL ENTER(JTIME,17,N,IQTY,NSKU)

```

```

05      C
06      C      IF REQUISITION IS TO REPLENISH A LOWER LEVEL,
07      C      SCHEDULE A RECEIVE EVENT(EVENT CODE 2).
08      C
09      JTIME=ITIME + LTPROD(NSKU)*YTDAY
10      IF((NSKU.NE.N).AND.(NSKU.LT.1000))
11      &      CALL ENTER(SZIME,2,NSKU,IOTYS,N)
12      C      UPDATE      PERFORMANCE
13      JOTYS=IOTYS
14      CALL CUM(ISHKPT,IOTYS,N)
15      CALL CUM(IFILLT,IOTYS,N)
16      C      IS THIS A PRIORITY 1 REQUISITION
17      IF(IPRY.NE.1) GO TO 85
18      C      UPDATE      PRIORITY 1 STATISTICS
19      CALL CUM(ISHIPT,IOTYS,N)
20      CALL CUM(IFILLI,IOTYS,N)
21      85 CONTINUE
22      RETURN
23      END

```

Subroutine: RIME, RIMEB

Function:

RIME is the main program for the Recoverable Item Management Evaluator simulation model. It is dimensioned to simulate up to 40 Stock Keeping Units. RIMEB is a "large-scale" version of RIME which permits simulation of up to 340 Stock Keeping Units.

Description:

This routine provides dimension information which establishes arrays sizes used in all other RIME routines. RIME is dimensioned to permit simulation of up to 40 SKUs, while RIMEB permits simulation of up to 340 SKUs.

RIME processing begins by reading the Exogenous Event File (EEF) identification record from logical file unit 07. This record defines the set of LRU/SRU families that were used to generate the EEF, and the number of bases, the number of quarters, and the number of replications that were used in the generation of this file.

RIME then reads the set of control cards from logical unit 05 which define output options which are to be utilized in the current simulation run and which also define the size of the simulation to be performed. See the Input Parameters Section, Volume I, for a detailed description of these variables.

The simulation process now begins. Subroutine ZERO is called to set the statistics arrays to zero, and subroutine INITM1 is called to read in item identification data for the first LRU/SRU group to be simulated. The replication loop then begins. Within this loop, subroutine EVNTS is called to simulate events for the current LRU/SRU group. When the simulation process is completed, and if the control flag ITWRT = 1, subroutine ITRSLT is called to punch cards containing 10 major statistics summarizing the simulation results for this replication of the current LRU/SRU group. The replication process then continues until all required replications for the current LRU/SRU group is completed. The next LRU/SRU group is then processed and this procedure continues until all required groups have been simulated.

Finally, the summary reports are produced. If ISUMRY = 1, subroutine OUTREP is called to produce a short-form summary of backorders, buy-dollars, and other inventory system results associated with set of control levels being evaluated. If IOUT is not equal to zero, subroutine OUT2 is called to produce a detailed statistical summary on a quarter by quarter basis.

The above simulation and reporting process then continues until all NLAM simulation runs have been completed.

01 10-20-79 14,569

[illegible]

01 10-20-79 15.565

```

3      COMMON/NEENAX/NEENAX
4      COMMON/NEIRST/NEIRST
5      COMMON/NETEN/NETEN
6      COMMON/NEOC/NEOC
7      COMMON/NEOCBK/NEOCBK
8      COMMON/NTIME/NTIME
9      C-----STIMULATION CONTROL PARAMETERS
0      COMMON/IDENT/IDENT
1      COMMON/NLAN/NLAN,NLAN
2      COMMON/NREPL/NREPL,NREPL
3      COMMON/NGROUP/NGROUP
4      COMMON/NFGRF/NFGRF
5      COMMON/NLGRF/NLGRF
6      C-----REPAIR SYSTEM VARIABLES
7      COMMON/NJOB/NJOB
8      COMMON/NBASES/NBASES
9      COMMON/NBRU/NBRU
0      COMMON/NBOTE/NBOTE
1      C-----
2      COMMON/INACT/INACT(40)
3      COMMON/INVDUS/INVDUS(40)
4      COMMON/INWIP/INWIP(40)
5      COMMON/ISBT/ISBT(40)
6      COMMON/IDBT/IDBT(40)
7      COMMON/IDONT/IDONT(40)
8      COMMON/IRL/IRL(40)
9      COMMON/IROL/IROL(40)
0      COMMON/IRQTY/IRQTY(40)
1      COMMON/ISUL/ISUL(40)
2      COMMON/ITL/ITL(40)
3      COMMON/LTADN/LTADN(40)
4      COMMON/LTPROD/LTPROD(40)
5      COMMON/NBOST/NBOST(40)
6      COMMON/NBOIU/NBOIU(40)
7      COMMON/NBOIX/NBOIX(40)
8      COMMON/NBOTE/NBOTE(40)
9      COMMON/NBOTU/NBOTU(40)
0      COMMON/NBENAS/NBENAS(40)
1      COMMON/NBETAC/NBETAC(40)
2      COMMON/NBEOAC/NBEOAC(40)
3      COMMON/NBENHC/NBENHC(1,20)
4      COMMON/NBETUR/NBETUR(1,20)
5      COMMON/NBEO/NBEO(1,20)
6      COMMON/NBENT/NBENT(40)
7      COMMON/NBOHAS/NBOHAS(1)
8      COMMON/NBOIS/NBOIS(1)
9      COMMON/NBOSUN/NBOSUN(1)
0      COMMON/NBOS /NBOS(1)
1      COMMON/NBOSN/NBOSN(1)
2      COMMON/NBROS/NBROS(1)
3      COMMON/NBTR /NBTR(1)
4      COMMON/NBGLT/NBGLT(1)

```

01 10-20-79 10.565

```

5      COMMON/XTREND/XTREND(1)
5      COMMON/UCOST/UCOST(40)
7      COMMON/XLOCPS/XLOCPS(500)
8      COMMON/JPSN/JPSN(500)
9      COMMON/JPOINTN/JPOINTN(500)
7      COMMON/JPRION/JPRION(500)
1      COMMON/JQTY/JQTY(500)
2      COMMON/JFINN/JFINN(500)
3      COMMON/JTYPN/JTYPN(500)
4      COMMON/IBACT/IBACT(200)
5      COMMON/IBFSN/IBFSN(200)
6      COMMON/IBCBK/IBCBK(200)
7      COMMON/IBRION/IBRION(200)
8      COMMON/IQTY/IQTY(200)
9      COMMON/ITNBA/ITNBA(200)
0      COMMON/IBOP/IBOPN(3),IBOPN(8)
1      COMMON/IBOPCT/IBOPN(3),IBOPN(8)
2      *****
3      C      PERFORMANCE STATISTICS
4      C
5      C
6      C
7      COMMON/IBACKI/IBACKI(16,3,6)
8      COMMON/IBACKI/IBACKI(16,3,6)
9      COMMON/IBACKI/IBACKI(16,3,6)
0      COMMON/IBACKI/IBACKI(16,3,6)
1      COMMON/IBODAT/IBODAT(16,3,6)
2      COMMON/IBODAT/IBODAT(16,3,6)
3      COMMON/IBANCL/IBANCL(16,3,6)
4      COMMON/IBISPS/IBISPS(16,3,6)
5      COMMON/IBXPS/IBXPS(16,3,6)
6      COMMON/IBILL/IBILL(16,3,6)
7      COMMON/IBILL/IBILL(16,3,6)
8      COMMON/IBVON/IBVON(16,3,6)
9      COMMON/IBVAT/IBVAT(16,3,6)
0      COMMON/IBVON/IBVON(16,3,6)
1      COMMON/IBORDN/IBORDN(16,3,6)
2      COMMON/IBATON/IBATON(16,3,6)
3      COMMON/IBECET/IBECET(16,3,6)
4      COMMON/IBOC /IBOC(16,3,6)
5      COMMON/IBOI /IBOI(16,3,6)
6      COMMON/IBOT/IBOT(16,3,6)
7      COMMON/IBTN/IBTN(16,3,6)
8      COMMON/IBNIP/IBNIP(16,3,6)
9      COMMON/IBNIP/IBNIP(16,3,6)
0      COMMON/IBNIP/IBNIP(16,3,6)
1      COMMON/IBNIP/IBNIP(16,3,6)
2      COMMON/IBNIP/IBNIP(16,3,6)
3      C-----
4      COMMON/IBNIP/IBNIP(16,3,6)
5      COMMON/IBNIP/IBNIP(16,3,6)
6      COMMON/IBNIP/IBNIP(16,3,6)

```

01 10-20-79 10.565

```

7      COMMON/EXECPL/EXECPL(16,3,8)
8      COMMON/INWP/INWP(16,3,6)
9      COMMON/INFP/INFP(16,3,6)
0      C-----C-----C-----C-----C-----C-----
1      C
2      COMMON/ETRAC/ETRAC,ISTRAC
3      C
4      C
5      C      REWIND EXOG. EVENT FILE
6      C
7      REWIND 67
8      C      READ INPUT DATA
9      C
0      10      CONTINUE
1      WRITE(6,8000)
2      8000      FORMAT(1H1, T30, 'INVENTORY SYSTEM SIMULATOR'//T34,
3                'RUN PARAMETERS'///)
4      C
5      C      READ EXOGENOUS EVENT FILE IDENTIFICATION RECORD
6      C
7      READ(7)NFORP,NLGRP,NBASES,NQTHR,NREPL
8      WRITE(6,8001)NFORP,NLGRP,NBASES,NQTHR,NREPL
9      8001      FORMAT(///" EXOGENOUS EVENT FILE CHARACTERISTICS"/
0                " NFORP",T16,"FIRST LRU GRP ="',T40,I5//
1                " NLGRP",T16,"LAST LRU GRP ="',T40,I5//
2                " NBASES",T16,"NUMBER OF BASES ="',T40,I5//
3                " NQTHR",T16,"NUMBER OF QUANTERS ="',T40,I5//
4                " NREPL",T16,"NUMBER OF REPLICATIONS ="',T40,I5//
5      C
6      C      READ OUTPUT CONTROLS(CNR CODE C2)
7      C
8      READ(5,8003)ITWRT,IOUT,IGRAPH,ISUMRY
9      8003      FORMAT(V)
0      WRITE(6,8004)ITWRT,IOUT,IGRAPH,ISUMRY
1      8004      FORMAT(///" (C2) OUTPUT CONTROLS...(NOTE. 10YES)"/
2                " ITWRT",T16,"IT.WRITE ="',I3//
3                " IOUT",T16,"SUMMARY ="',I3//
4                " IGRAPH",T16,"GRAPHS ="',I3//
5                " ISUMRY",T16,"SUMMARY ="',I3//
6      C
7      C      READ DEBUG FLAGS
8      C
9      READ(5,8005)IDBUG,ISBUG,IFBUG,ISBUG,INBUG,ITRAC,ISTRAC
0      WRITE(6,8006)IDBUG,ISBUG,IFBUG,ISBUG,INBUG,ITRAC,ISTRAC
1      8005      FORMAT(///" (C3) DEBUG FLAGS"/
2                T16,"IDBUG ="',I3//
3                T16,"ISBUG ="',I3//
4                T16,"IFBUG ="',I3//
5                T16,"ISBUG ="',I3//
6                T16,"INBUG ="',I3//
7                T16,"ITRAC ="',I3//
8                T16,"ISTRAC ="',I3//)

```

01 10-20-79 10.565

```

9      C
10     C      DEFINE ITEM INPUT FILES
11     C
12     C
13     C      READ SIMULATION SIZE PARAMETERS(CARD TYPE C8)
14     C
15     READ(5,8003)NLAN,INOTR,NTOTL
16     WRITE(6,8080)NLAN,INOTR,NTOTL
17     8080  FORMAT(///' (C8) SIMULATION SIZE'///
18           &      ' NLAN',T16,'NUMBER OF LAMBDA'S',T40,I5//
19           &      ' INOTR',T16,'NUMBER OF QUANTERS',T40,I5//
20           &      ' NTOTL',T16,'NO. OF LEU GROUPS',T40,I5//
21           &      )
22     C
23     C      LIMIT INPUT PARAMETERS CONSISTENT WITH EXD6. FILE
24     C
25     IF(INOTR.GT.NNOTR)INOTR=NNOTR
26     K=NLCRP+NFGRP+1
27     IF(NTOTR.GT.K)NTOTL=K
28     C
29     C---INITIALIZE RANDOM NUMBER STREAM
30     C
31     RNLAST=RANDU(0.1)
32     C
33     C      REWIND THE LEVELS FILE (09)
34     C
35     REWIND 09
36     C
37     C*****
38     C
39     C---BEGIN LAGRANGIAN LOOP
40     C
41     DO 200 NLAN=1,NLAN
42     IF(IDBUG.EQ.1)WRITE(6,8090)NLAN
43     C
44     8090  FORMAT(//10('*****'),'BEGIN SIMULATION ',
45           &      'FOR NLAN=',I4,10('*****'))//
46     C
47     C
48     C      REWIND THE EVENTS FILE
49     C      AND READ THE HEADER RECORD AGAIN
50     C
51     REWIND 07
52     READ(7)NFCRP,NLCRP,NBASES,NNOTR,NREPL
53     C
54     C
55     C      ZERO THE STATISTICS ARRAYS
56     C
57     CALL ZERO
58     C
59     100 CONTINUE
60     C

```

01 10-20-79 10.369

```

1      C      SIMULATE ALL STOTL GROUPS. ONE GROUP AT A TIME
2      C
3      NITEM=1
4      NGROUP=NGRPS +1
5      DO 400 NM1,STOTL
6      NGROUP=NGROUP+1
7      WRITE(6,213)N
8      213  FORMAT('*****BEGIN LRU GRP=#,NS')
9      C
10     C---READ DATA FOR GROUP N
11     C
12     CALL INETH1
13     C
14     C---BEGIN REPLICATION LOOP
15     C
16     DO 390 KREPL,1,KREPL
17     C
18     C
19     C
20     C---SIMULATE THIS ITEM
21     C
22     CALL RYNTS(RNLAST)
23     C
24     C---RECORD RANDOM NUMBER SEED
25     C
26     RNLAST=RANDU(RNLAST)
27     IF(ITWET.NO.3)CALL ITSELT
28     C
29     C      END OF REPLICATION LOOP(KREPL)
30     390 CONTINUE
31     C
32     C      END OF GROUP LOOP
33     400 CONTINUE
34     C
35     C
36     C
37     C-----OUTPUT PERFORMANCE STATISTICS
38     C      NOTE---OUTREP PARAMETER IS J (ACTIONS,UNITS,DOLLARS)
39     C      ---OUT2 PARAMETER IS KRSKU(BASE,DEPOT,OVERHAUL)
40     C      FOR SRU AND LRU.
41     C
42     IF(ISUMRY,EG.1) CALL OUTREP(4,2)
43     IF(IGUT,NN.0) CALL OUT2(1,6)
44     C-----END OF LAGRANGIAN LOOP
45     2000 CONTINUE
46     C
47     C---END OF JOB
48     9999 CONTINUE
49     STOP
50     END
W      7 MEMORY EXPANDED. USE $LIMITS OR $CORE= OPTION FOR NEXT RUN

```

```

1  *#RUF=;RIME/OBJ/RIMEB,0(BCD;NOGO)
2  *RIMEB,S
3  CHARACTER TEXT*40
4  C
5  C      SET DIMENSIONS FOR STATISTICS ARRAYS
6  C
7  C
8  C-----DEFINE GASP COMMONS-----
9  C
10  DIMENSION NSET(1)
11  COMMON QSET(600)
12  EQUIVALENCE(NSET(1),QSET(1))
13  COMMON /GCOM1/ ATRIE(30),JEVNT,MFA,MFE(100),MLE(100),HSTOP,HCRDU,
14  A  NNABO,NNAPT,NNATP,NNFEL,MNC(100),NTRY,NPRNE,PPARM(50,4),THOW,TTBEG,
15  A  TTCLN,TTFIN,TTTIB(80),TTSET
16  COMMON /GCOM2/ EENQ(100),TINN(100),KKRNE(400),MHAXQ(100),
17  A  QUTIN(100),SSOBV(25,5),SSTPV(25,6),VVNC(100)
18  C
19  *-----
20  COMMON/C2CNT/ITWET,IOUT,IGRAPH,ISUMRY
21  COMMON/IDBUG/IDBUG
22  COMMON/IEBUG/IEBUG
23  COMMON/IGBUG/IGBUG
24  COMMON/IPBUG/IPBUG
25  COMMON/INBUG/INBUG
26  COMMON/IDDIV/IDDIV
27  COMMON/IDLEV/IDLEV
28  COMMON/IDSTAT/IDSTAT
29  COMMON/IDTHQ/IDTHQ
30  COMMON/IKDEM/IKDEM
31  COMMON/INQTR/INQTR
32  COMMON/IPFOR/IPFOR
33  COMMON/IQTRND/IQTRND
34  COMMON/ISTAT/ISTAT
35  COMMON/ISTOCK/ISTOCK,IDSTOC
36  COMMON/ISTOP/ISTOP
37  COMMON/ITCANS/ITCANS,ITCANR
38  COMMON/ITDAY/ITDAY
39  COMMON/ITDIV/ITDIV
40  COMMON/ITFOR/ITFOR,IDFOR
41  COMMON/ITHQ/ITHQ
42  COMMON/ITINV/ITINV
43  COMMON/ITIME/ITIME
44  COMMON/ITLEV/ITLEV
45  COMMON/ITMNT/ITMNT
46  COMMON/ITQTR/ITQTR
47  COMMON/ITWEEK/ITWEEK
48  COMMON/ITYEAR/ITYEAR
49  COMMON/NSMAX/NSMAX
50  COMMON/NDEM/NDEM
51  COMMON/NDHIS/NDHIS
52  COMMON/NENTRY/NENTRY

```

RIMEB

```

53      COMMON/NFEMAX/NFEMAX
54      COMMON/NFIFIRST/NFIFIRST
55      COMMON/NITEM/NITEM
56      COMMON/NLOC/NLOC
57      COMMON/NLOCBK/NLOCBK
58      COMMON/NTIME/NTIME
59      C-----SIMULATION CONTROL PARAMETERS
60      COMMON/IDENT/IDENT
61      COMMON/BLAM/BLAM,MLAM
62      COMMON/NREPL/NREPL,NREPL
63      COMMON/NGROUP/NGROUP
64      COMMON/NFGRP /NFGRP
65      COMMON/NLGRP /NLGRP
66      C-----REPAIR SYSTEM VARIABLES
67      COMMON/NJOB/NJOB
68      COMMON/NBASES/NBASES
69      COMMON/NSRU/NSRU
70      COMMON/NNQTR/NNQTR
71      C-----
72      COMMON/INVACT/INVACT(340)
73      COMMON/INVDUE/INVDUE(340)
74      COMMON/INWIP/INWIP(340)
75      COMMON/IBRT/IBRT(340)
76      COMMON/IDBT/IDBT(340)
77      COMMON/IDORT/IDORT(340)
78      COMMON/IRL/IRL(340)
79      COMMON/IROL/IROL(340)
80      COMMON/IRQTY/IRQTY(340)
81      COMMON/ISUL/ISUL(340)
82      COMMON/ITL/ITL(340)
83      COMMON/LTADM/LTADM(340)
84      COMMON/LTPROD/LTPROD(340)
85      COMMON/NBOPT/NBOPT(340)
86      COMMON/NBOIU/NBOIU(340)
87      COMMON/NBOIS/NBOIS(340)
88      COMMON/NBOTS/NBOTS(340)
89      COMMON/NBOTU/NBOTU(340)
90      COMMON/NDEHAC/NDEHAC(340)
91      COMMON/NRETAC/NRETAC(340)
92      COMMON/NREQAC/NREQAC(340)
93      COMMON/NDEHND/NDEHND(1,24)
94      COMMON/NRETUR/NRETUR(1,24)
95      COMMON/NREQO/NREQO(1,24)
96      COMMON/NDENT/NDENT(340)
97      COMMON/REQHAD/REQHAD(1)
98      COMMON/REQSIZ/REQSIZ(1)
99      COMMON/REXSUM/REXSUM(1)
00      COMMON/RHAD /RHAD(1)
01      COMMON/RHEEN/RHEEN(1)
02      COMMON/RHREQS/RHREQS(1)
03      COMMON/RHTBR /RHTBR(1)
04      COMMON/RSIGLT/RSIGLT(1)

```

```

05 COMMON/RTREND/PTREND(4)
06 COMMON/UCOST/UCOST(200)
07 COMMON/ILOCPE/ILOCPE(500)
08 COMMON/JPSN/JPSN(500)
09 COMMON/JPOINT/JPOINT(500)
10 COMMON/IPRIOR/IPRIOR(500)
11 COMMON/JQTY/JQTY(500)
12 COMMON/JTIME/JTIME(500)
13 COMMON/JTYPE/JTYPE(500)
14 COMMON/IBACPT/IBACPT(500)
15 COMMON/IDFSNB/IDFSNB(500)
16 COMMON/ILOCBK/ILOCBK(500)
17 COMMON/IPRIOR/IPRIOR(500)
18 COMMON/IQTYB/IQTYB(500)
19 COMMON/ITMBAC/ITMBAC(500)
20 COMMON/IBOP/IBOP(3),IBOPOR(3)
21 COMMON/IBOPCT/IBOPSH(3),IBOPLS(3)
22 *****
23 C PERFORMANCE STATISTICS
24 C
25 C
26 C
27 COMMON/IBACKI/IBACKI(16,3,6)
28 COMMON/IBACKT/IBACKT(16,3,6)
29 COMMON/IBAKDI/IBAKDI(16,3,6)
30 COMMON/IBAKDT/IBAKDT(16,3,6)
31 COMMON/IBODAT/IBODAT(16,3,6)
32 COMMON/IBODAI/IBODAI(16,3,6)
33 COMMON/ICANCL/ICANCL(16,3,6)
34 COMMON/IDISPS/IDISPS(16,3,6)
35 COMMON/IDISPD/IDISPD(16,3,6)
36 COMMON/IFILLI/IFILLI(16,3,6)
37 COMMON/IFILLT/IFILLT(16,3,6)
38 COMMON/INVOH/INVOH(16,3,6)
39 COMMON/INVDAY/INVDAT(16,3,6)
40 COMMON/INVOR/INVOR(16,3,6)
41 COMMON/IORDER/IORDER(16,3,6)
42 COMMON/IRATON/IRATON(16,3,6)
43 COMMON/IRECET/IRECET(16,3,6)
44 COMMON/IREOC /IREOC(16,3,6)
45 COMMON/IREOT /IREOT(16,3,6)
46 COMMON/IREOT/IREOT(16,3,6)
47 COMMON/IRETRN/IRETRN(16,3,6)
48 COMMON/ISHIPT/ISHIPT(16,3,6)
49 COMMON/ISHIPT/ISHIPT(16,3,6)
50 COMMON/ITERM /ITERM(16,3,6)
51 COMMON/ISHORD/ISHORD(16,3,6)
52 COMMON/ILGORD/ILGORD(16,3,6)
53 C-----
54 COMMON/IREPGN/IREPGN(16,3,6)
55 COMMON/INRTS/INRTS(16,3,6)
56 COMMON/ICNDEN/ICNDEN(16,3,6)

```

57		COMMON/IRECPL/IRECPL(16,3,6)	110
58		COMMON/IWIP/IWIP(16,3,6)	
59		COMMON/IWPP/IWPP(16,3,6)	
60		C-----	
61	C		
62		COMMON/ITRACE/ITRACE,ISTRAC	
63	C		
64	C		
65	C	NEWEND EXOG. EVENT FILE	
66	C		
67		REWIND 07	
68	C	READ INPUT DATA	
69	C		
70	10	CONTINUE	
71		WRITE(6,8000)	
72	8000	FORMAT(1H1, T40,'INVENTORY SYSTEM SIMULATOR'//T34,	
73	&	'RUN PARAMETERS'///)	
74	C		
75	C	READ EXOGENOUS EVENT FILE IDENTIFICATION RECORD	
76	C		
77		READ(7)NFGRP,NLGRP,NBASES,NNOTE,NREPL	
78		WRITE(6,8001)NFGRP,NLGRP,NBASES,NNOTE,NREPL	
79	8001	FORMAT(///' EXOGENOUS EVENT FILE CHARACTERISTICS'//	
80	&	" NFGRP",T16,"FIRST LEU GRP =",T40,I5//	
81	&	" NLGRP",T16,"LAST LEU GRP =",T40,I5//	
82	&	" NBASES",T16,"NUMBER OF BASES =",T40,I5//	
83	&	" NNOTE",T16,"NUMBER OF QUARTERS =",T40,I5//	
84	&	" NREPL",T16,"NUMBER OF REPLICATIONS",T40,I5)	
85	C		
86	C	READ OUTPUT CONTROLS(CARD CODE C2)	
87	C		
88		READ(5,8003)ITWRT,IOUT,IGRAPH,ISUMRY	
89	8003	FORMAT(V)	
90		WRITE(6,8004)ITWRT,IOUT,IGRAPH,ISUMRY	
91	8004	FORMAT(///' (C2) OUTPUT CONTROLS...(NOTE, 1=YES)'//	
92	&	' ITWRT',T16,'IT,WRITE'=',I3//	
93	&	' IOUT',T16,'SUMMARY ='',I3//	
94	&	' IGRAPH',T16,'GRAPHS #',I3//	
95	&	' ISUMRY',T16,'SUMMARY #',I3)	
96	C		
97	C	READ DEBUG FLAG	
98	C		
99		READ(5,8003)IDBUG,IEBUG,IFBUG,IGBUG,INEUG,ITRACE,ISTRAC	
00		WRITE(6,8005)IDBUG,IEBUG,IFBUG,IGBUG,INEUG,ITRACE,ISTRAC	
01	8005	FORMAT(///' (C3) DEBUG FLAG'//	
02	&	T16,'IDBUG ='',I3//	
03	&	T16,'IEBUG ='',I3//	
04	&	T16,'IFBUG ='',I3//	
05	&	T16,'IGBUG ='',I3//	
06	&	T16,'INEUG ='',I3//	
07	&	T16,'ITRACE'=',I3//	
08	&	T16,'ISTRAC'=',I3//)	

```

109      C
110      C          DEFINE ITEM INPUT FILES          111
111      C
112      C
113      C          READ SIMULATION SIZE PARAMETERS (CARD TYPE C8)
114      C
115      C          READ(5,8003)NLAN,INQTR,NTOTL
116      C          WRITE(8,8080)NLAN,INQTR,NTOTL
117      C          8080      FORMAT(///' (C8) SIMULATION SIZE'///
118      C          C          ' NLAN',T16,'NUMBER OF LAMBDA',T40,I5//
119      C          C          ' INQTR',T16,'NUMBER OF QUARTERS',T40,I5//
120      C          C          ' NTOTL',T16,'NO. OF LRU GROUPS',T40,I5//
121      C          C          )
122      C
123      C          LIMIT INPUT PARAMETERS CONSISTENT WITH EXOG. FILE
124      C
125      C          IF(INQTR.GT.NNQTR)INQTR=NNQTR
126      C          K=NLRGP+NFGRP+1
127      C          IF(NTOTL.GT.K)NTOTL=K
128      C
129      C---INITIALIZE RANDOM NUMBER STREAM
130      C
131      C          RNLAST=RANDU(7,1)
132      C
133      C          REWIND THE LEVELS FILE (09)
134      C
135      C          REWIND 09
136      C
137      C*****
138      C
139      C---BEGIN LAGRANGIAN LOOP
140      C
141      C          DO 2000 NLAN=1,NLAN
142      C          IF(IDBUG.EQ.1)WRITE(8,8090)NLAN
143      C
144      C          8090 FORMAT(//10('*****'),'BEGIN SIMULATION ',
145      C          C          "FOR NLAN=",I4,10('*****')//)
146      C
147      C
148      C          REWIND THE EVENTS FILE
149      C          AND READ THE HEADER RECORD AGAIN
150      C
151      C          REWIND 07
152      C          READ(7)NFGRP,NLRGP,NBASES,NNQTR,NPEPL
153      C
154      C
155      C          4PRO THE STATISTICS ARRAYS
156      C
157      C          CALL ZERO
158      C
159      C          100 CONTINUE
160      C

```

```

61      C      SIMULATE ALL NTOTL GROUPS, ONE GRUPE AT A TIME
62      C
63      NITEM=1
64      NGROUP=NGRUP -1
65      DO 400 N=1,NTOTL
66      NGROUP=NGROUP+1
67      WRITE(6,213)N
68      213  FORMAT('***BEGIN LRU GRP=',I5)
69      C
70      C---READ DATA FOR GROUP N
71      C
72      CALL INTRM1
73      C
74      C---BEGIN REPLICATION LOOP
75      C
76      DO 390 KREPL=1,NREPL
77      C
78      C
79      C
80      C---SIMULATE THIS ITEM
81      C
82      CALL EVNTS(RNLAST)
83      C
84      C---RECORD RANDOM NUMBER SEED
85      C
86      RNLAST=RANDU(RNLAST)
87      IF(ITWRT.EC,1)CALL ITRSLT
88      C
89      C      END OF REPLICATION LOOP(KREPL)
90      390  CONTINUE
91      C
92      C      END OF GROUP LOOP
93      400  CONTINUE
94      C
95      C
96      C
97      C-----OUTPUT PERFORMANCE STATISTICS
98      C      NOTE---OUTREP PARAMETER IS J (ACTIONS,UNITS,DOLLARS)
99      C      ---OUT2  PARAMETER IS KNSKU(BASE,DEPOT,OVERHAUL,
100      C      FOR SRU AND LRU.
101      C
102      IF(IISUMRY.EC,1) CALL OUTREP(1,3)
103      IF(IOUT.NE,0) CALL OUT2(1,5)
104      C-----END OF LAGRANGIAN LOOP
105      2000 CONTINUE
106      C
107      C---END OF JOB
108      9999  CONTINUE
109      STOP
110      END
111      *M      7 MEMORY EXPANDED, USE SLYNITS OR CORE= OPTION FOR NEXT RUN

```

Subroutine: REVIEW**Functions**

This routine compares the inventory position = (on-hand + on-order + work-in-process - backorders) to the desired stock level (IROL(N)). If the inventory position is below this level, an order for the deficiency is placed.

Calling Parameters

NN = Item review variable. If NN = 0, all items are reviewed.
Otherwise, only item NN is reviewed.

IFLAG = Initial provisioning flag. If IFLAG = 1, this routine schedules all required orders for immediate receipt to simulate initial provisioning stock positioning. Otherwise, replenishment orders are scheduled for receipt after the standard administrative and production leadtimes.

Description:

The routine first computes the inventory position equal to (stock on-hand + due-in + in-process less backorders) for Stock Keeping Unit N. If the inventory position is less than the reorder level (IROL(N)) for SKU N, a replenishment order is initiated. If IFLAG equals 1, routine ORDERV is called. This routine schedules

delivery of the required units to occur at time $ITIME + 1$ to simulate initial provisioning and positioning activities. Otherwise, subroutine ORDER is called to schedule the replenishment order. In this latter case, delivery time is computed by subroutine ORDER to occur after the standard administrative and production leadtimes.

01 10-20-79 10.250 JS

```

1  *NRUN=INTIME/ONS/REVIEW.O(NCD,NOWOY
2  *REVIEW,S
3  SUBROUTINE REVIEW(NN,IPLAGY
4  C      THIS ROUTINE COMPARES ONE INVENTORY POSITION=
5  C      (ON-HAND + ON-ORDER + WORK-IN-PROCESS - BACKORDERS)
6  C      TO THE DESIRED STOCK LEVEL (INOL(N)).
7  C      IF THE INVENTORY POSITION IS BELOW THIS LEVEL,
8  C      AN ORDER FOR THE DEFICIENCY IS PLACED?
9  C
10 C      IF NN=0, ALL ITEMS ARE REVIEWED. OTHERWISE, ONLY
11 C      ITEM NN IS REVIEWED.
12 C
13 C
14 C      IF IPLAG=1, SCHEDULES FOR IMMEDIATE RECEIPT TO
15 C      SIMULATE INSTANT PROVISIONING STOCK POSITIONS.
16 C      OTHERWISE, SCHEDULE ORDERS FOR RECEIPT AFTER THE
17 C      STANDARD RDN AND FROM LEADTIME.
18 C      FOR STANDARD DELIVERIES, DELIVERY TIMES ARE COMBINED
19 C      BY SUBROUTINE ORDER.
20 C
21 C
22 COMMON/EDBUS/EDBUS
23 COMMON/ICDFOR/ICDFOR
24 COMMON/ITIME/ITIME
25 COMMON/ITMTR/ITMTR
26 COMMON/EDMAX/EDMAX
27 COMMON/XTM/XTM
28 COMMON/MLCCK/MLCCK
29 COMMON/INVACT/INVACT(1)
30 COMMON/INVDUS/INVDUS(1)
31 COMMON/INWIP/INWIP(1)
32 COMMON/INOL/INOL(1)
33 COMMON/INQTY/INQTY(1)
34 COMMON/LTADN/LTADN(1)
35 COMMON/LTPROB/LTPROB(1)
36 COMMON/UBOTU/UBOTU(1)
37 COMMON/UBOTE/UBOTE(1)
38 C
39 C      IF NN=0, REVIEW ALL ITEMS; OTHERWISE, ONLY REVIEW ITEM NN
40 C
41 C      NP=NN
42 C      NL=NN
43 C      IF(NN.EQ.0)NP=1
44 C      IF(NN.EQ.0)NL=XTM
45 C
46 C      DO 2000 NUN=NP,NL
47 C      N=NN
48 C
49 C
50 C      NRS=UBOTE(N)
51 C      INPOS=INVACT(N)+INVDUS(N)+INWIP(N)-UBOTU(N)
52 C      IF(EDBUS,50,1)WRITE (6,5015)N,INVACT(N),INVDUS(N),

```

REVIEW

03T 01 10-20-79 10.850 JS

```

53      *      INVTG(N),INVTG(N),INVTG(N),INVTG(N)
54      8013 FORMAT(4X,"*****REVIEW*****" ON=","ISP" DUE-IN=","IS,
55      *      " INVTG,"IS," SO=","ISP" INVTG,"IS," SO-SHORT=","IS.
56      *      " INVTG,"IS)
57      C
58      C      COMPARE STOCK POSITION TO REORDER POINT
59      C
60      IF(INVTG(N).INVTG(N)) GO TO 2000
61      *
62      *      COMPUTE STOCK LEVEL DEFICIENCY
63      *
64      97 CONTINUE
65      INVTG(N)=INVTG
66      *
67      *      COMPUTE DELIVERY DATE FOR ORDER
68      *
69      95 IF(IFLG.NE.1) GO TO 120
70      *
71      *      PLACE ORDER FOR INVTG UNITS
72      C      DELIVERY TIME IS COMPUTED BY SUBROUTINE ORDER,
73      *
74      CALL ORDER(N,INVTG,0)
75      GO TO 2000
76      C
77      C      SCHEDULE INITIAL PROVISIONING SHIPMENTS FOR
78      C      IMMEDIATE DELIVERY FROM THE VENDOR
79      120 CONTINUE
80      IDELVTIME+1
81      CALL ORDER(N,INVTG,IDELV)
82      2000 CONTINUE
83      RETURN
84      END

```

Subroutine: SSTAT**Function:**

Event Type 11. This routine updates time-persistent and end-of-period statistics.

Calling Parameters:

IP3 = The week number associated with the previous call to subroutine SSTAT

Description:

This routine is called at the end of each simulation week. Subroutine SSTAT first updates the performance statistics INVDAY, IBAKDT, and IBAKDI. These variables record inventory - week and backorder week measures.

At the end of each quarter, the routine computes values for the end-of-period variables INVOH, INVOR, IBACKT, and IBACKI. These variables record the end-of-period status for on hand stocks, on order stocks, total backorders, and priority 1 backorders, respectively. The period counter KEND (which denotes the week number within the quarter) and ITINV (the current quarter number) are also incremented by 1 at the end of each quarter. Finally, the routine ends by calling subroutine ENTER to schedule a new type 11 event.

```

1  PRUN=TIME/ONJ/SSTAT.O(BCD,NOGO)
2  PSSTAT,S SPECIAL STATISTICS ROUTINE
3  SUBROUTINE SSTAT(IP3)
4      C SUBROUTINE SSTAT
5      COMMON/IDSTAT/IDSTAT
6      COMMON/ISTAT/ISTAT
7      COMMON/ISTOP/ISTOP
8      COMMON/ITIME/ITIME
9      COMMON/ITINV/ITINV
10     COMMON/ITQTR/ITQTR
11     COMMON/NITEM/NITEM
12     COMMON/NBMAX/NBMAX
13     COMMON/NLOCBK/NLOCBK
14     COMMON/INVACT/INVACT(1)
15     COMMON/INVDUE/INVDUE(1)
16     COMMON/INWIP/INWIP(1)
17     COMMON/NDENT/NDENT(1)
18     COMMON/IBACKI/IBACKI(16,3,6)
19     COMMON/IBACKT/IBACKT(16,3,6)
20     COMMON/IBAKDT/IBAKDT(16,3,6)
21     COMMON/IBAKDI/IBAKDI(16,3,6)
22     COMMON/INVDAY/INVDAY(16,3,6)
23     COMMON/INVOH/INVOH(16,3,6)
24     COMMON/INVOR/INVOR(16,3,6)
25     COMMON/INIP/INIP(16,3,6)
26     COMMON/NBOTU/NBOTU(1)
27     COMMON/NBOTR/NBOTR(1)
28     COMMON/NBOIR/NBOIR(1)
29     COMMON/NBCIN/NBCIN(1)
30     COMMON/UCOST/UCOST(1)
31
32     C
33     I = ITINV
34     KEND=IP3
35
36     C
37     DO 50 NN=1,NITEM
38     M=NN
39     KK=KNSKU(M)
40     IF(INVACT(M).LE.0)GO TO 20
41
42     C UPDATE ON-HAND INVENTORY-WEEKS COUNTERS
43     CALL CUM(INVDAY,INVACT(M),M)
44     CONTINUE
45
46     C IF THERE ARE NO BACKORDERS FOR ITEM M.
47     GO TO 50
48
49     C IF(NBOTU(M).LE.0) GO TO 50
50
51     C UPDATE BACKORDER STATISTICS
52     IBAKDT(I,1,KK)=IBAKDT(I,1,KK)+NBOTR(M)

```

SSTAT

```

53      CALL CUMB(IBAKDT,NBOTU(N),N)
54      IF(NBOTU(N).LE.0) GO TO 50
55      IBAKDI(I,1,KK)=IBAKDI(I,1,KK)+NBOTR(N)
56      CALL CUMB(IBAKDI,NBOTU(N),N)
57      50      CONTINUE
58      60      CONTINUE
59      C
60      KB = ITQTR/IDSTAT
61      IF(KEND,GE,KB)GO TO 100
62      KEND = KEND + 1
63      GO TO 200
64      C
65      C      UPDATE END OF PERIOD STATISTICS
66      C
67      100      CONTINUE
68      DO 150 NN=1,NITEM
69      N=NN
70      KK=KNSKU(N)
71      IF(INWIP(N).GT.0)CALL CUM(INWIP,INWIP(N),N)
72      IF(INVACT(N).GT.0) CALL CUM(INVCH,INVACT(N),N)
73      IF(INVDUE(N).GT.0) CALL CUM(INVOR,INVDUE(N),N)
74      C
75      C      UPDATE EOP BACKORDER STATISTICS
76      C
77      IF(NBOTU(N).LE.0) GO TO 150
78      IBACKT(I,1,KK)=IBACKT(I,1,KK)+NBOTR(N)
79      CALL CUMB(IBACKT,NBOTU(N),N)
80      IF(NBOTU(N).LE.0) GO TO 150
81      IBACKI(I,1,KK)=IBACKI(I,1,KK)+NBOTR(N)
82      CALL CUMB(IBACKI,NBOTU(N),N)
83      150      CONTINUE
84      160      CONTINUE
85      C
86      C      INCREMENT PERIOD COUNTERS
87      C
88      KEND=1
89      ITINV=ITINV+1
90      C
91      C      CREATE NEXT STATISTICS COLLECTION EVENT
92      C
93      200      CONTINUE
94      ISTAT = ISTAT + IDSTAT
95      CALL ENTER(ISTAT,11,KEND,KEND,KEND)
96      RETURN
97      END

```

Subroutine: STATHD

Function:

This routine is a utility routine called by subroutine OUT2. The routine zeros the totals vector ITOTL, and writes a heading for the summary statistics report.

Calling Parameters:

J = Index identifying the statistics header information to be printed

ITOTL = The statistics vector to be set to zero

1T 01 10-20-79 10.871

```

1  SUBROUTINE STATHD(J,ITOTL)
2      C      PRINT STATISTIC MEASURE FOR PARTICULAR REPORT(J,1,2,OR 3)
3      DIMENSION ITOTL(17)
4      C
5      C      ZERO TOTALS ARRAY
6      C
7      DO 5 I=1,29
8          ITOTL(I)=0
9      5  CONTINUE
10     C
11     GO TO(10,20,30),J
12     10 WRITE(6,104)
13     GO TO 30
14     20 WRITE(6,106)
15     GO TO 30
16     30 WRITE(6,107)
17     104 FORMAT(//.T56,"*** ACTIONS/NSM A ***")
18     107 FORMAT(//.T56,"$$$ DOLLARS $$$")
19     106 FORMAT(//.T56,"... UNITS ...")
20     40 CONTINUE
21     RETURN
22     END

```

STATHD

Subroutine: WRIFEL

Function:

This routine writes to File 06 all of the current elements in the future events list.

Description:

This routine provides a detailed listing of all information currently recorded in the Future Events List. See Volume I, Section II for a detailed description of this routine.

```
1      SUBROUTINE WRIFEL
2      COMMON/NFIRST/NFIRST
3      COMMON/NENTRY/NENTRY
4      COMMON/JPOINT/JPOINT(1)
5      COMMON/JPSN/JPSN(1)
6      COMMON/JQTY/JQTY(1)
7      COMMON/JPRIOR/JPRIOR(1)
8      COMMON/JTYPE/JTYPE(1)
9      COMMON/JTIME/JTIME(1)
10     COMMON/ILOCPE/ILOCPE(1)
11     WRITE(6,23)JTIME(NFIRST),NFIRST,NENTRY
12     23 FORMAT(/30X,"FUTURE EVENTS LIST AT TIME ",I10/
13     &      40X,"NFIRST=",I10," NENTRY=",I10//
14     &      T20,"K      JTIME      JTYPE      JPSN      JQTY      JPRIOR",
15     &      "      JPOINT      INDEX"//)
16     C
17     K=0
18     KK=NFIRST
19     10 CONTINUE
20     IF(KK,LE.0) GO TO 900
21     K=K+1
22     WRITE(6,43)K,JTIME(KK),JTYPE(KK),JPSN(KK),JQTY(KK),
23     &      JPRIOR(KK),JPOINT(KK),KK
24     43 FORMAT(T10,8I10)
25     KK=JPOINT(KK)
26     GO TO 10
27     900 CONTINUE
28     WRITE(6,903)
29     903 FORMAT(/T30,"NO MORE ENTRIES ON THE FUTURE EVENTS LIST")
30     RETURN
31     END
```

Subroutine: ZERO

Function:

This routine zeros the RIME statistical accumulators.

Description:

This routine is called at the beginning of each RIME run to zero the variables for recording performance statistics, beginning on hand and on order stocks.

```

1  *#RUN=;RTH/ONE/ZERO.O.W(BCB,NOBODY
2  *ZERO,S  ZERO STATISTICAL ACCUMULATORS
3      SUBROUTINE ZERO
4      COMMON/IOBLIS/IOBLIS
5      COMMON/IBOP/IBOP(3),IBOP(8)
6      COMMON/IBAUTN/IBAUTN(12)
7      COMMON/IBPROJ/IBPROJ(12)
8      COMMON/IBACKI/IBACKI(16,3,6)
9      COMMON/IBACKT/IBACKT(16,3,6)
10     COMMON/IBAKDI/IBAKDI(16,3,6)
11     COMMON/IBAKDT/IBAKDT(16,3,6)
12     COMMON/IBODAT/IBODAT(16,3,6)
13     COMMON/IBODAT/IBODAT(16,3,6)
14     COMMON/ICANCL/ICANCL(16,3,6)
15     COMMON/IXISPS/IXISPS(16,3,6)
16     COMMON/IXEPSB/IXEPSB(16,3,6)
17     COMMON/IXILLI/IXILLI(16,3,6)
18     COMMON/IXILLT/IXILLT(16,3,6)
19     COMMON/INVDAT/INVDAT(16,3,6)
20     COMMON/INVOB/INVOB(16,3,6)
21     COMMON/INVOB/INVOB(16,3,6)
22     COMMON/IORDER/IORDER(16,3,6)
23     COMMON/INATON/INATON(16,3,6)
24     COMMON/IRECET/IRECET(16,3,6)
25     COMMON/IREOC /IREOC(16,3,6)
26     COMMON/IREOI /IREOI(16,3,6)
27     COMMON/IREOT/IREOT(16,3,6)
28     COMMON/IRETRN/IRETRN(16,3,6)
29     COMMON/ISHIPX/ISHIPX(16,3,6)
30     COMMON/ISHIPT/ISHIPT(16,3,6)
31     COMMON/ITERN /ITERN(16,3,6)
32     COMMON/IGNORD/IGNORD(16,3,6)
33     COMMON/ILCORB/ILCORB(16,3,6)
34     COMMON/IREPCB/IREPCB(16,3,6)
35     COMMON/INETS/INETS(16,3,6)
36     COMMON/ICUDEN/ICUDEN(16,3,6)
37     COMMON/IRECPL/IRECPL(16,3,6)
38     COMMON/INWP/INWP(16,3,6)
39     COMMON/INWP/INWP(16,3,6)

```

C

COMMON/IBOPCT/IBOPCT(3),IBOPCT(3)

C

ZERO STATISTICAL ACCUMULATORS

```

13  IOBLIS=0
14  DO 25 I=1,16
15  DO 25 J=1,3
16  DO 25 K=1,6
17  ICANCL(I,J,K)=0
18  INVOB(I,J,K)=0
19  INVOB(I,J,K)=0
20  IRECT(I,J,K)=0
21  IRETRN(I,J,K)=0
22  INVDAT(I,J,K)=0

```

ZERO

01 10-20-79 10.561

ZERO STATISTICAL ACCUMULATORS

```

3  YORDER(I,J,K)=0
4  IDISPS(I,J,K)=0
5  ITEN(I,J,K)=0
6  IEXPND(I,J,K)=0
7  INATON(I,J,K)=0
8  IRROC(I,J,K)=0
9  IRNOT(I,J,K)=0
10 IRNOI(I,J,K)=0
11 IBACKT(I,J,K)=0
12 IBACKI(I,J,K)=0
13 IBAKUT(I,J,K)=0
14 IBARDI(I,J,K)=0
15 IBODXT(I,J,K)=0
16 IBODXI(I,J,K)=0
17 IFILTI(I,J,K)=0
18 IFILZI(I,J,K)=0
19 ISHIFT(I,J,K)=0
20 ISHIZI(I,J,K)=0
21 ISHOND(I,J,K)=0
22 ILORD(I,J,K)=0
23 IENPEN(I,J,K)=0
24 IURTS(I,J,K)=0
25 IENBEN(I,J,K)=0
26 INECP(I,J,K)=0
27 IWI(I,J,K)=0
28 IWP(I,J,K)=0
29 25 CONTINUE

```

C
C
C
C

ZERO ON-HAND AND ON-ORDER COUNTERS

```

30 DO 110 I=1,3
31  INOPON(I)=0
32  IBOPON(I)=0
33  INOPSH(I)=0
34  IBOPLO(I)=0
35 110 CONTINUE
36  RETURN
37  END

```

REFERENCES

1. Demmy, W. Steven, RIME: The Recoverable Item Management Evaluator, Volume I: Model Description, TR-80-01, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 153 pp.
2. Demmy, W. Steven, RIME: The Recoverable Item Management Evaluation Volume II, Program Listings and Narratives, TR-80-02, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 298 pp.
3. Demmy, W. Steven, An Empirical Evaluation of Proposed Stockage Policies for Recoverable Item Management, TR-80-03, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 173 pp.
4. Demmy, W. Steven and Victor J. Presutti, Jr., Multi-Echelon Inventory Theory in the Air Force Logistics Command, Working Paper 76-3011-27, Department of Management, Wright State University, Dayton, Ohio 45435.
5. Muckstadt, John A., "A Model for a Multi-Item, Multi-Echelon, Multi-Indenture Inventory System, Management Science, v20, n4, December, 1973, pp. 472-481.