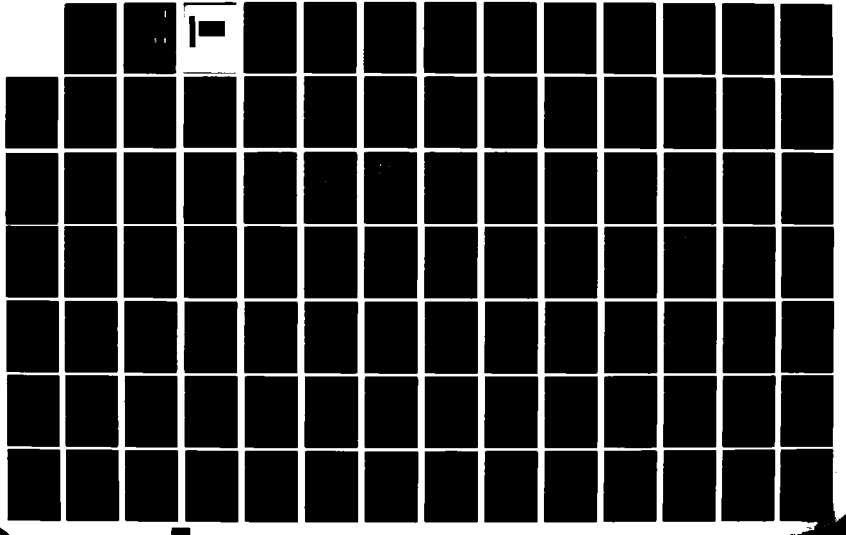
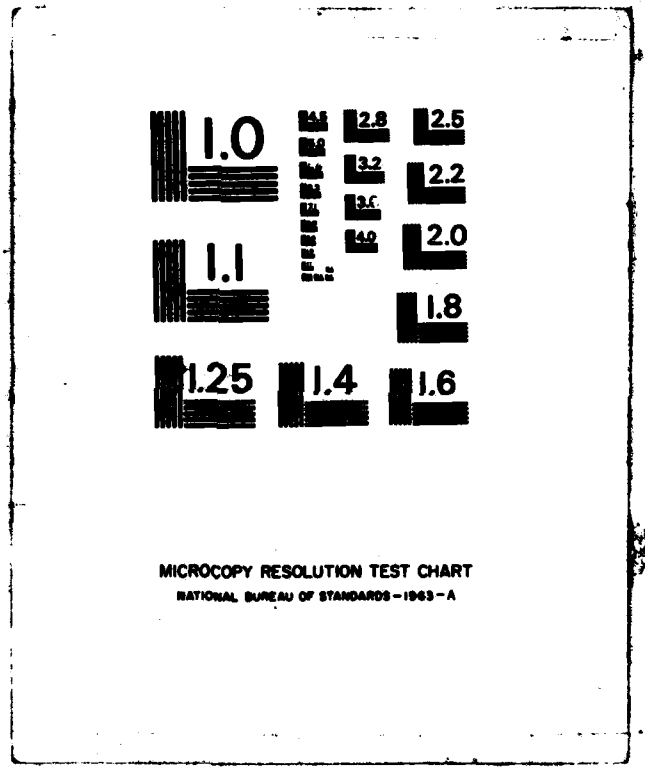
AD-A121 987 RIME: THE RECOVERABLE ITEM MANAGEMENT EVALUATOR VOLUME II SECTION III RIM.. (U) DECISION SYSTEMS DAYTON OH W S DEMMY MAY 80 TR-80-02-C F33600-78-C-0524 1/2

UNCLASSIFIED F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

PHOTOGRAPH THIS SHEET

AD A121987

DTIC ACCESSION NUMBER

III

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
S DEC 1 1982 D
D

DATE ACCESSIONED

DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

DTIC FORM OCT 79 70A

DOCUMENT PROCESSING SHEET

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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

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

6. PERFORMING ORG. REPORT NUMBER
TR-80-02-C

7. AUTHOR(s)
W. Steven Demmy

8. CONTRACT OR GRANT NUMBER(s)
C-0524
F33600-78-~~D-0314~~

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Decision Systems
3575 Charlene Drive
Dayton, Ohio 45432

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
23041A5

11. CONTROLLING OFFICE NAME AND ADDRESS
2750th ABW/PMA
BLDG 1, Area C
Wright-Patterson AFB, Ohio

12. REPORT DATE
May 1980

13. NUMBER OF PAGES
298

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

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 interations 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	Program
BWAIT	NRTS
CONDEM	ORDER
CUM, CUMB	OUT2
ENTER	OUTREP
ENTERB	RCVPRT
EVNTS	RECEIV
FILLBO	REQ
	REMOVE
FILLST	RIME
INGASP	RIMEB
INFEL	REVIEW
INITAL	REPGEN
	REQ
INITEM, INITM1, INITM2	SSTAT
ITRSLT	STATHD
KNSKU	WRIFEL
LEVEL	ZERO
LEVEL2	

Subroutine: BWAIT

Function:

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.

```
      SUBROUTINE BWAIT(N,NNEED,NJOB)
C                      EVENT CODE 16. THIS ROUTINE SIMULATES AN LRU ENTERI
C                   WAIT-FOR-PARTS STATUS.
C
      COMMON /GCOM1/ ATRIB(30),JEVNT,MFA,MFE(100),MLE(100),MSTOP,NCRDR,
     & NNAPO,NNAPT,NNATR,NNFIL,NNQ(100),NNTRY,NPRNT,PPARM(50,4),TNOW,TTBEG,
     &  TTCLR,TTFIN,TTRIB(20),TTSET
      COMMON /GCOM6/ EENQ(100),IINN(100),KKRNK(100),MMAXQ(100),
     &  QQTIM(100),SSOBV(25,5),SSTPV(25,6),VVNC(100)
      DIMENSION NSET(1)
      COMMON QSET(1)
      EQUIVALENCE (NSET(1),QSET(1))
      COMMON /ITIME /ITIME
C
C          FILE NJOB IN THE WAIT-FOR-PARTS FILE
C
      ATRIB(1) = N
      ATRIB(2) = NNEED
      ATRIB(3) = NJOB
      ATRIB(4) = ITIME
      CALL FILEM (2)
      RETURN
      END
```

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.

```
      SUBROUTINE CONDEM(N,IQTY,NJOB)
C      EVENT CODE 15.  THIS ROUTINE RECORDS THE CONDEMNATION
C      OF IQTY UNITS OF SKU N. NJOB IS THE ASSOCIATED
C      REP GEN NUMBER.
C
      COMMON/ICNDEM/ICNDEM(16,3,6)
      COMMON/INWIP/INWIP(1)
C
C      UPDATE CONDEMNATION STATUS
C
      CALL CUM(ICNDEM,IQTY,N)
C
C      UPDATE WORK-IN-PROCESS
C
      INWIP(N)=INWIP(N)-IQTY
C
C
      RETURN
     END
```

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

```
*#RUN=&RIMS/08J/CUM.O(BCD,NOGO)
*CUM.S       CUMULATE SIMULATION STATISTICS
            SUBROUTINE CUM(ISTAT,IQTY,KFSN)
C                    THIS ROUTINE UPDATES THE ITEMS, UNITS, AND DOLLARS
C                     ASSOCIATED WITH THE STATISTIC ISTAT
C
      DIMENSION ISTAT(16,3,6)
      COMMON/ITINV/ITINV
      COMMON/UCOST/UCOST(1)
      COMMON/IGBUG/IGBUG
C
      I=ITINV
C
C                  COMPUTE K (STATISTICS COLECTION INDEX),WHERE
C                        K=1 DENOTES LRU AT BASE
C                        K=2 DENOTES SRU AT BASE
C                        K=3 DENOTES LRU AT DEPOT
C                        K=4 DENOTES SRU AT DEPOT
C                        K=5 DENOTES LRU AT OVERHAUL
C                        K=6 DENOTES SRU AT OVERHAUL
C
      K=KNSKU(KFSN)
C
C                  UPDATE COUNT OF FSN ACTIONS
C
      IF (IQTY.GT.0)GO TO 21
      ISTAT (I,1,K)=ISTAT (I,1,K)+1
      GO TO 22
   21 CONTINUE
      ISTAT (I,1,K)=ISTAT(I,1,K)+1
      GO TO 22
C
C                   UPDATE UNIT AND DOLLAR COUNTS, NOTE THAT ENTRY
C                   POINT CUMB HAS NO EFFECT ON FSN ACTION COUNTS.
C
      ENTRY CUMB(ISTAT,IQTY,KFSN)
    I=ITINV
      K=KNSKU(KFSN)
   22  ISTAT (I,2,K)=ISTAT (I,2,K)+IQTY
      DOLLAR=UCOST(KFSN)*FLOAT(IQTY)
      ISTAT(I,3,K)=ISTAT(I,3,K)+IFIX(DOLLAR+.5)
C
      IF(IGBUG.NE.1)RETURN
      WRITE(6,23)KFSN,IQTY,I,K,(ISTAT(I,J,K),J=1,3)
   23 FORMAT(A(5"*"*"),"CUM-@KFSN=",I5," IQTY=",
     &  I5," ITINV=",I5," K=",I5," UPDATED STATS=",3I8)
C
      RETURN
    END
```

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.

```
@RUN=,RIME/OBJ/FELIST.O(BCD,NOGO)
@FELIST,S
      SUBROUTINE ENTER(KTIME,KTYPE,KFSN,KQTY,KPRIOR)
C        THIS     ROUTINE ENTERS EVENTS ON THE FUTURE EVENTS LIST
C        AND UPDATES THE CHAIN STRUCTURE
      COMMON/IDBUG/IDBUG
      COMMON/ITIME/ITIME
      COMMON/NENTRY/NENTRY
      COMMON/NFEMAX/NFEMAX
      COMMON/NFIRST/NFIRST
      COMMON/NLOC/NLOC
      COMMON/NTIME/NTIME
      COMMON/ILOCFE/ILOCFE(1)
      COMMON/JFSN/JFSN(1)
      COMMON/JPOINT/JPOINT(1)
      COMMON/JPRIOR/JPRIOR(1)
      COMMON/JQTY/JQTY(1)
      COMMON/JTIME/JTIME(1)
      COMMON/JTYPE/JTYPE(1)
C
C           CHECK IF EVENT TIME EXCEEDS 16 QTRS(134400 TIME UNITS).
C                 IF SO, GO TO STATEMENT 100 AND RETURN.
C
      IF(KTIME.GT.134400) GO TO 100
C
   10 CONTINUE
C           UPDATE NO. OF ENTRIES ON THE F.E.L
C
      NENTRY=NENTRY+1
C      IF THIS ENTRY EXCEEDS CAPACITY OF THE CHAIN,PRINT
C      ERROR MESSAGE AND QUIT
      IF(NENTRY.GT.NFEMAX) GO TO 90
C      PLACE TRANSACTION IN FIRST AVAILABLE LOCATION
   20 K=ILOCFE(NENTRY)
      NLOC=K
      JTIME(K)=KTIME
      JTYPE(K)=KTYPE
      JFSN(K)=KFSN
      JQTY(K)=KQTY
      JPRIOR(K)=KPRIOR
C      CHECK TO SEE IF THIS IS THE ONLY ENTRY ON THE LIST
      IF(NENTRY.NE.1) GO TO 40
C      UPDATE       LIST STATUS AND      POINTER VARIABLES
      NFIRST=K
      NTIME=KTIME
      JPOINT(K)=0
      GO TO 100
C      DOES THE NEW TRANSACTION PRECEED ALL OTHER LIST ENTRIES(
   40 IF(KTIME.GE.NTIME) GO TO      60
C      INSERT       NEW TRANSACTION      AS A FIRST LINK IN THE CHAIN
      NTIME=KTIME
      JPOINT(K)=NFIRST
```

```
T 01  10.20.79   10.421   .8
```

```
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 JJ=NFIRST
58              KJ=JPOINT(JJ)
59        C      IS JJ THE LAST RECORD ON THE CHAIN(
60           62 IF(KJ.EQ.0) GO TO 97
61        C      DOES THE NEW TRANSACTION PRECEED KJ(
62              IF(KTIME.LT.JTIME(KJ)) GO TO 80
63              JJ=KJ
64              KJ=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)NFEMAX
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,KPRIOR,
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 MOD (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 orginated 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 (LPRI - IPRI)/100. If the parameter LPRI is greater than 1000, the backorder represents a requisition for SRU components needed to compiete the repair of an LRU. In this case, the job number, NJOB, associated with the LRU repair is given by NJOB = (LPRI - 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       $RUN*;RIME/OBJ/ENTERB,O(BGD,NOGO)
2             SUBROUTINE ENTERB(N,IQTY,LPRI,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 LPRI > 1000, THEN LPRI=1000*MJOR + 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(1)
8              COMMON/IBACPT/IBACPT(1)
9              COMMON/IDPSNB/IDPSNB(1)
0              COMMON/ILOCBK/ILOCBK(1)
1              COMMON/IPRIOR/IPRIOR(1)
2              COMMON/IQTYB/IQTYB(1)
3              COMMON/ITMBAC/ITMBAC(1)
4       C           COMPUTE PRIORITY AND STOCK KEEPING UNIT
5       C
6            NSKU=N
7            IPRI=MOD(LPRI,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)
```

```
      91 FORMAT(//20X,"ERROR-BACKORDER FILE OVERFLOW, ")
         WRITE(6,82)
      82 FORMAT(1H1,10X,23H**BACKORDER FILE DUMP**)
         DO 83 K=1,NBMAX
      83 WRITE(6,84) K,ITMBAC(K),IDFSNB(K),IPRIOR(K),IQTYB(K),IBACPT(K)
      84 FORMAT(//3X,"REC NO=",I3,5X,"ITMBAC=",I7,5X,"IDFSNB=",I10,5X,
     &   7HPRIOR=,I1,5X,6HIQTYB=,I7,5X,7HIBACPT=,I3)
         STOP
     C   RECORD QUANTITY,PRIORITY,FSN ID,AND TIME DATA FOR THIS BO REQ
      20 ITMBAC(IPT)=JTIME
         IDFSNB(IPT)=NSKU
         IPRIOR(IPT)=IPRI
         IQTYB(IPT)=IQTY
     C    ARE ANY OTHER BACKORDERS OUTSTANDING ON ITEM N
         IF(NBOPT(N).GT.0) GO TO 40
     C    RECORD      POINTER DATA
         NBOPT(N)=IPT
         IBACPT(IPT)=0
         RETURN
     C    IS THE      NEW BO A PRIORITY 1 REQUISITION
      40 IF(IPRI.EQ.1) GO TO 60
     C    NOTE-       NEW LOW PRIORITY BACKORDERS ARE INSERTED LAST ON THE
     C        BACKORDER CHAIN, THE REMAINING STEPS IN THIS PORTION
     C        OF THE SUBROUTINE ACCOMPLISH THIS OBJECTIVE
     C    SET JPT EQUAL TO THE FILE LOCATION NO OF THE FIRST BACKORDERED
     C    REQUISITION IN THE CHAIN
         JPT=NBOPT(N)
     C    IS JPT      THE LAST LINK IN THE CHAIN
      49 IF(IBACPT(JPT).EQ.0) GO TO 50
         KPT=IBACPT(JPT)
         JPT=KPT
         GO TO 49
     C    RECORD      NEW POINTERS
      50 IBACPT(JPT)=IPT
         IBACPT(IPT)=0
         RETURN
     C    SET JPT EQUAL TO LOCATION NO OF FIRST BO ON CHAIN
      60 JPT=NBOPT(N)
     C    IS BACKORDER JPT A HIGH PRI BO
      61 IF(IPRIOR(JPT).NE.1) GO TO 80
     C    IS JPT      THE LAST BACKORDER ON THE CHAIN
         IF(IBACPT(JPT).EQ.0) GO TO 62
         KPT=JPT
         JPT=IBACPT(KPT)
         GO TO 61
     C    INSERT      NEW BO AS LAST LINK ON CHAIN
      62 IBACPT(JPT)=IPT
         IBACPT(IPT)=0
         RETURN
     C    IS JPT      THE ONLY BO ON CHAIN
      80 IF(JPT.NE.NBOPT(N)) GO TO 81
     C    INSERT      NEW BO AS FIRST      LINK ON CHAIN
```

```
1-04  10-20-79   10.294
15              NBOPT(N)=IPT
16              IBACPT(IPT)=JPT
17              RETURN
18        C      INSERT     NEW BO AS LINK BETWEEN KPT AND JPT
19           81 IBACPT(KPT)=KPT
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 appropirate event subroutine logic is then repeated until a type 10 event is eventually encoutered.

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 Exogeneous 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.

```
*#RUN*JTIME/OBJ/EVNTS,O(BCD,NOGO)
*EVNTS.S
      SUBROUTINE EVNTS(RNLAST)
C           THIS ROUTINE TAKES EVENTS OFF THE FUTURE EVENTS LIST
C              AND THE EXOGENOUS EVENT FILE BASED ON LOWEST EVENT
C              TIME. IT THEN CALLS THE APPROPRIATE EVENT SUBROUTINE.
C
      COMMON/ETIME/ETIME
      COMMON/NTIME/NTIME
      COMMON/IDBUG/IDBUG
      COMMON/ITRACE/ITRACE,ISTRAC
      COMMON/NITEM/NITEM
       COMMON/INVACT/INVACT(1)
       COMMON/INVDUE/INVDUE(1)
       COMMON/INWIP/INWIP(1)
       COMMON/NBOTR/NBOTR(1)
       COMMON/NBOXR/NBOXR(1)
       COMMON/NBOTU/NBOTU(1)
       COMMON/NBOXU/NBOXU(1)
C
      COMMON/NREPL/NREPL
C
C***********************************
C
C           INITIALIZE RANDOM NUMBER GENERATOR
C
       X=-RNLAST
       CALL RANDU(X)
    15 CONTINUE
C             INITIALIZE THE FUTURE EVENTS LIST(INITAL).
C
       CALL INITAL
C
C           INITIALIZE DATA ARRAYS FOR THIS ITEM
C
      CALL INITM2
C
C           READ FIRST ENTRY ON EXOG. EVENT FILE
CC
       READ(7)JTIME,JTYPE,JP3,JP4,JP5
C
       IF(IDBUG      .NE. 1) GO TO 16
       CALL WRIFEL
    16 CONTINUE
C
C**************************************************
C
   100 CONTINUE
C
C           IF NEXT EVENT IS EXOG., GO TO 150
C
      IF(JTIME.LT.NTIME) GO TO 150
```

```
C
C              ENDOGENOUS EVENT.   REMOVE IT FROM THE FEL.
C
           CALL REMOVE(ITIME,ITYPE,IP3,IP4,IP5)
           GO TO 160
C
C              EXOGENOUS EVENT.  SET CURRENT EVENT PARAMETERS,
C                    THEN READ NEXT EXOG. EVENT FROM FILE 7.
C
       150 CONTINUE
           ITIME=JTIME
           ITYPE=JTYPE
           IP3=JP3
           IP4=JP4
           IP5=JP5
            READ(7,END=155)JTIME,JTYPE,JP3,JP4,JP5
           GO TO 160
C
C              SET JTIME TO INDICATE THERE ARE NO MORE EXOG. EVENTS.
       155 CONTINUE
           JTIME=999999999
      160 CONTINUE
           IF(IDBUG.EQ.1)WRITE(6,151)
       151 FORMAT(" ITIME ITYPE",T20,"N",T26,"QTY",T35,"IP5",
     &             T63,"N",T49,"INVACT INVDUE INWIP  NBOIU  NBOTU",
     &          T85,"NBOIR  NBOTR  NTIME  JTIME")
           IF(IDBUG.EQ.1)
     &             WRITE(6,153)ITIME,ITYPE,IP3,IP4,IP5,NTIME,JTIME
       153 FORMAT(1X,I7,3I8,T98,2I7)
C
C
            GO TO ( 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
     &       35, 36, 37, 38, 39, 40),ITYPE
C
C*********************************************************
C
        21 CONTINUE
C            REFLECT RECEIPT OF TRANSACTION
C            REQUISITION
C
            CALL REQ(IP3,IP4,IP5,ITIME)
           CALL REVIEW(IP3,0)
            GO TO 90
        22 CONTINUE
C            RECEIPT OF SHIPMENT
C            REFLECT RECEIPT
            CALL RECEIV(IP3,IP4)
            GO TO 90
        23 CONTINUE
                CALL CANCLB(IP3,IP4,IP5)
            GO TO 90
C
```

```
C              SERVICEABLE RETURN EVENT
C
      24      CALL RET(IP3,IP4,ITIME)
           GO TO 90
      25 CONTINUE
C          STATUS       REVIEW
         CALL REVIEW(0,0)
         IF(IP5.LE.0) GO TO 90
           MTIME=ITIME+IP5
         CALL ENTER(MTIME,5,0,0,IP5)
           GO TO 90
      26 CONTINUE
C          LEVELS       COMPUTATION
         CALL LEVEL(0)
         CALL REVIEW(0,0)
         IF(IP5.LE.0) GO TO 90
           MTIME=ITIME+IP5
         CALL ENTER(MTIME,6,0,0,IP5)
           GO TO 90
      27 CONTINUE
C          BUY GUIDELINE
           CALL GUIDE
           GO TO 90
      28 CONTINUE
C          BUDGET       REVIEW
           CALL BUDGET
           GO TO 90
      29 CONTINUE
C          FORECAST UPDATE
           CALL FORUPD
           GO TO 90
      30 CONTINUE
C************************************
C
C          END OF       RUN
C
C************************************
C
C              READ EXOGENOUS EVENT FILE TILL A TYPE 10 EVENT
C                IS ENCOUNTERED, MARKING THE END OF THE CURRENT
C               LRU/SRU GROUP DATA SET.
C
         IF(JTYPE.EQ.10)GO TO 99
         READ(7)JTIME,JTYPE,JP3,JP4,JP5
         IF(IDEBUG.EQ.1)WRITE(6,8310)JTIME,JTYPE,JP3,JP4,JP5
    8310 FORMAT(" NEXT EXOG. EVENT=",5I6)
         GO TO 30
C
      31 CONTINUE
C          SPECIAL STATISTICS
           CALL SSTAT(IP3)
           GO TO 90
```

```
   32 CONTINUE
C          DEMAND       PARAMETER UPDATE
      CALL DEMPAR(IP3,IP4,IP5)
      GO TO 90
   33 CONTINUE
C             EVENT 13.  UPDATE TRACE PARAMETERS
      IF(ITIME.EQ.ITRACE)IDBUG=1
      IF(ITIME.EQ.ISTRAC)IDBUG=0
      GO TO 90
C
C
C
C             CODE 14.  REPARABLE GENERATION
C
   34 CONTINUE
      CALL REPGEN(IP3,IP4,IP5)
      GO TO 90
C
C             CODE 15. CONDEMNATION
C
   35  CONTINUE
      CALL CONDEM(IP3,IP4,IP5)
      CALL REVIEW(IP3,0)
      GO TO 90
C
C             CODE 16. BEGIN WRITING FOR PARTS
C
   36  CONTINUE
      CALL NWRIT(IP3,IP4,IP5)
      GO TO 90
C
C
C             CODE 17.  RECEIVE PARTS
C
   37  CONTINUE
      CALL RCVPRT(IP3,IP4,IP5)
      GO TO 90
C
C              CODE 18. REPAIR COMPLETION
C
   38  CONTINUE
      CALL CREPR(IP3,IP4,IP5)
      GO TO 90
C
C             CODE 19.  NRTS EVENT
C
   39 CONTINUE
      CALL NRTS(IP3,IP4,IP5)
      CALL REVIEW(IP3,0)
      GO TO 90
C
C
```

```
      C
      C                          EVENT 20.  INITIAL PROVISIONING.
   40 CONTINUE
      C            INITIAL PROVISIONING EVENT.  ORDER FOR IMMEDIATE DELIVERY
      C                      SUFFICIENT STOCK TO BRING EACH LOCATION UP TO ITS
      C                      STOCK LEVEL.
      C
         CALL LEVEL(0)
         CALL REVIEW(0,1)
   90 CONTINUE
      C
      C
      C**********PRINT BO STATUS
         IF(IDBUG.NE.1) GO TO 180
        IF((ITYPE.GE.5).AND.(ITYPE.LE.18))GO TO 180
        N=IP3
        WRITE(6,2993)N,INVACT(N),INVDUE(N),INWEP(N),NBOXU(N),NBOTU(N),
      &               NBOXR(N),NBOTR(N)
  2993 FORMAT(T89,2I8,6I7)
      C
  180 CONTINUE
      C
         IF(IDBUG.EQ.1)PRINT 93,ITIME,ITYPE,IP3,IP4,IP5,
      &             (INVACT(N),INVDUE(N),INWIP(N),NBOTU(N),N=1,NITEM)
   93 FORMAT(I7,3I3,I7,(T24,8(3I5,I2X"*")))
      C
         GO TO 100
      C
      C
   99 CONTINUE
          RETURN
          END
```

Subroutine: FILLBO

Function:

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.

```
*#RUN=ARIMS/CBS/FILLBO.O(BCD,NOGO)
*FILLBO.S
      SUBROUTINE FILLBO(N)
      COMMON/IDBUG/IDBUG
      COMMON/NBOPT/NBOPT(1)
      COMMON/INVACT/INVACT(1)
      COMMON/IQTYB/IQTYB(1)
      COMMON/ITMBAC/ITMBAC(1)
      COMMON/ISUL/ISUL(1)
      COMMON/IDFSNB/IDFSNB(1)
      COMMON/IPRIOR/IPRIOR(1)
      COMMON/IBACPT/IBKCPT(1)
      COMMON/NLOCBK/NLOCBK
      COMMON/ILOCBK/ILOCBK(1)
   25 IF(NBOPT(N).LE.0) RETURN
      IF(IDBUG.NE.1) GO TO 27
        I=NBOPT(N)
        WRITE(6,990)N,INVACT(N),IQTYB(I),IPRIOR(I),ITMBAC(I),I
  990         FORMAT(' ****FILLBO-NSKU=',I5,'  ONH=',I5,'  IQTYB=',I5,
     &        1  IPRIOR=',I5,'  ITMBAC=',I8,'  IPT=',I5)
   27 CONTINUE
C           SET IPT EQUAL TO THE POINTER NUMBER OF THE OLDEST,HIGHEST
C           PRIORITY REQUISITION FOR ITEM N ON BACKORDER STATUS.
      IPT=NBOPT(N)
       NSKU=IDFSNB(IPT)
        IQTYS=IQTYB(IPT)
        IPRI=IPRIOR(IPT)
      ITMBO=ITMBAC(IPT)
        IF(INVACT(N).LE.0)RETURN
C     **** NOTE ****
C        (A) THIS ROUTINE ASSUMES BO REQUISITIONS ARE FILLED USING
C            A FIFO,BY PRIORITY,ISSUE RULE
C        (B) IF ONHAND ASSETS ARE INSUFFICIENT TO COMPLETELY FILL A
C            GIVEN REQUISITION,PARTIAL SHIPMENTS ARE INITIATED.
C        (C) REDUCED SHIPMENTS TO PARTIALLY FULFILL SEVERAL
C            OUTSTANDING BACKORDERS ARE NOT PERMITTED IN THIS ROUTINE
C
C           SET ITEST EQUAL TO THE        REMAINING ON-HAND INVENTORY,
C           IF THE        REQUISITION WAS         TO BE COMPLETELY FILLED
        ITEST=INVACT(N)-IQTYS
C           IS ITEST ABOVE THE SUPPORT LEVEL
      IF(ITEST.GE.ISUL(N)) GO TO 80
C           IS THIS A PRIORITY 1 REQUISITION
      IF(IPRI.NE.1) GO TO 60
C
C          PRIORITY 1 REQ
C
C           CAN THE REQ BE COMPLETELY SATISFIED FROM ON HAND STOCK
      IF(INVACT(N).GE.IQTYS) GO TO 80
C           SHIP ALL REMAINING ON-HAND STOCK IN PARTIAL FULFILLMENT
C           OF THIS REQUISITION
      IQTYS=INVACT(N)
```

```
            GO TO 80
      C
      C           PRIORITY IS BACKORDER
      C
      C           ARE ON-HAND ASSETS ABOVE THE SUPPORT LEVEL
         60 IF(INVACT(N).LE.ISUL(N))          GO TO 70
      C           SHIP DOWN TO THE SUPPORT LEVEL
            IQTYS=INVACT(N)-ISUL(N)
      C           UPDATE        STOCK STATUS RECORDS TO REFLECT THE PARTIAL SHIP
        65           CONTINUE
           CALL FELLST(N,IQTYS,IPRI,ITENO,0,NSKU)
            IQTYB(IPT)=IQTYB(IPT)-IQTYS
      C           AT THIS POINT,NO FURTHER SHIPMENTS SHOULD BE POSSIBLE, SINCE
      C           HIGH PRIORITY REQUISITIONS ARE PROCESSED FIRST
         70 RETURN
      C           SHIP TO FILL THE ENTIRE REQUISITION
         80 IQTYS=IQTYB(IPT)
      C           UPDATE        STOCK STATUS RECORDS
           CALL FELLST(N,IQTYS,IPRI,ITENO,1,NSKU)
      C           REMOVE        REQUISITION FROM THE BACKORDER FILE
        100 NBOPT(N)=IBAKPT(IPT)
            NLOCBK=NLOCBK+1
            ILOCBK(NLOCBK)=IPT
      C           RETURN        TO BEGINNING OF        ROUTINE TO SEE IF ANY MOR
      C           NEED TO BE PROCESSED.
            GO TO 25
            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.

```
      SUBROUTINE FILLST(N,IQTY,IPRI,ITMBO,NR,NSKU)
C
C     UPDATE BACKORDER STATISTICS TO REFLECT SHIPMENT OF IQTY
C UNITS OF ITEM N TO SATISFY A PRIORITY I BACKORDER.
C IF NR=1, THE NUMBER OF REQUISITIONS BACKORDERED IS REDUCED BY 1
C OTHERWISE, NR=0, REFLECTING A SHIPMENT TO PARTIALLY FILL
C THE CURRENT BACKORDER.
C
C           IF NSKU.GE. 1000, SCHEDULE A RECEIVE PARTS
C             EVENT (EVENT CODE 17)
C
       COMMON/ITIME/ITIME
       COMMON/ITDAY/ITDAY
      COMMON/IBODAT/IBODAT(1)
      COMMON/IBODAI/IBODAI(1)
       COMMON/LTPROD/LTPROD(1)
         COMMON/IDBUG/IDBUG
         COMMON/ISHIPT/ISHIPT(1)
         COMMON/ISHIPI/ISHIPI(1)
         COMMON/NBOTU/NBOTU(1)
         COMMON/NBOIU/NBOIU(1)
         COMMON/NBOTR/NBOTR(1)
         COMMON/NBOIR/NBOIR(1)
         COMMON/INVACT/INVACT(1)
C
C           UPDATE BACKORDER TIME STATISTICS
C
      IBOTM=ITIME-ITMBO
      IUNDAY=IBOTM*IQTY
C
      CALL CUM(IBODAT,IUNDAY,N)
      IF(IPRI.EQ.1)CALL CUM(IBODAI,IUNDAY,N)
C
C
C             IF NSKU > 1000, SCHEDULE A RECEIVE PARTS
C                EVENT(EVENT CODE 17).
C
      JTIME= ITIME + 10
      IF(NSKU.GE.1000) CALL ENTER(JTIME,17,N,IQTY,NSKU)
C
C              IF REQUISITION IS TO REPLENISH A LOWER
C                   SUPPLY LEVEL, SCHEDULE A RECEIVE EVENT.
C
      JTIME= ITIME + LTPROD(NSKU)*ITDAY
      IF((NSKU.GT.1000).AND.(N.NE.NSKU))
     &           CALL ENTER(JTIME,1,NSKU,IQTY,N)
C
C          REDUCE ON-HAND STOCKS AND BACKORDER STATUS
C
         INVACT(N)=INVACT(N)-IQTY
         NBOTU(N)=NBOTU(N)-IQTY
         NBOTR(N)=NBOTR(N)-NR
```

```
      CALL CUM(ISHIPT,IOTY,N)
      IF(ISRE.NE.1)GO TO 20
      NBOIU(N)=NBOIU(N)-IOTY
      NBOIR(N)=NBOIR(N)-NR
      CALL CUM(ISHIPT,IOTY,N)
   20    CONTINUE
      IF(IDBUG.NE.1)RETURN
      WRITE(6,23)N,NSKU,IOTY,ISRE,ITMBAC,NR,NBOTU(N),NBOIU(N),
     &   NBOTR(N),NBOIR(N)
   23 FORMAT(10X," ***FILLST  N="I4,"   NSKU=",I5,
     &   "   IOTY=",I5,
     &  '  ISRE=',I5,'  ITMBAC=',I8,'  NR=',I5/
     & T20,"NBOTU=",I5,"  NBOIU=",I5,"  NBOTR=",I5,
     & "  NBOIR=",I5)
      RETURN
      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.

```
      SUBROUTINE INFEL
C     THIS ROUTINE INITIALIZES THE FUTURE EVENTS
C     LIST TO AN EMPTY STATUS
      COMMON/NENTRY/NENTRY
      COMMON/NFEMAX/NFEMAX
      COMMON/NTIME/NTIME
      COMMON/XLOCFE/XLOCFE(1)
      NENTRY=0
      NTIME=9999999
      DO 1 I=1,NFEMAX
    1 XLOCFE(I)=I
      RETURN
      END
```

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 NNTRY, 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.

```
*#RUN*=;RIME/OBJ/INGASP.O(BCD,NOGO)
*INGASP,S
      SUBROUTINE INGASP(KTRY,KFIL,KSET,KATR)
       COMMON /GCOM1/ ATRIB(30),JEVNT,MFA,MFE(100),MLE(100),MSTOP,NCRDR,
     & NNAPO,NNAPT,NNATR,NNFIL,NNQ(100),NNTRY,NPRNT,PPARM(50,4),TNOW,TTBEG,
     &  TTCLR,TTFIN,TTRIB(30),TTSET
       COMMON /GCOM6/ EENQ(100),IINN(100),KKRNK(100),MMAXQ(100),
     &  QQTIM(100),SSOBV(25,5),SSTPV(25,6),VVNQ(100)
C
      COMMON/IDBUG/IDBUG
      COMMON/ITIME/ITIME
      MAXMIN=KTRY*(KATR+2)
      IF(IDBUG.EQ.1)WRITE(6,13)ITIME,KTRY,KFIL,KSET,KATR,MAXMIN
   13 FORMAT("*****INGASP--ITIME=",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)
C
C
C          *INITIALIZE GASP FILE SYSTEM**
C
C      SET FILE SYSTEM PARAMETERS                    6.50
C    FIRST,SET MAX FILE ENTRIES
C
      NNTRY=KTRY
C
C             SET NUMER OF FILE
C
      NNFIL=KFIL
C
C              SET DEMINSION OF NSET
C
      NNSET=KSET
C
C              SET NUMBER OF ATTRIBUTES
C
      NNATR=KATR
            NNAPT=NNATR+2
            NNAPO=NNATR+1
C
C
C           SET TIME PARAMETER
C
            TNOW=0
C
C
C        SET DEFAULT FILE STRUCTURE BASED ON LOW VALUE
C              FIRST IN ATTRIBUTE 3
C
      DO 50 IFL=1,NNFIL
       KKRNK(IFL)=3
```

```
      IINN(IPL)=1
   50 CONTINUE
C
C                INITIALIZE FILE SYSTEM POINTERS
C
             CALL SET
C
C**                 END OF GASP INITIALIZATION**                *
      RETURN
      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.

```
*#RUN=IRIME/OBJ/INITAL.O,W(BCD,NOBO)
*INITAL.S
      SUBROUTINE INITAL
C          THIS ROUTINE INITIALIZES THE FUTURE EVENTS LIST AND RECORDER
C           FILES,  SETS THE TIMING PARAMETERS USED BY THIS SIMULATION;
C           AND SCHEDULES INITIAL ENDOGENOUS EVENTS.
C
     COMMON/IDBUG/IDBUG
      COMMON/ITRACE/ITRACE,ISTRAC
      COMMON/IDDIV/IDDIV
      COMMON/IDLEVL/IDLEVL
      COMMON/IDSTAT/IDSTAT
      COMMON/IDTHQ/IDTHQ
      COMMON/IKDEM/IKDEM
      COMMON/INQTR/INQTR
      COMMON/IQTRND/IQTRND
      COMMON/ISTOCK/ISTOCK,IDSTOC
      COMMON/ISTOP/ISTOP
      COMMON/ISTAT/ISTAT
      COMMON/ITCANB/ITCANB,IDCANB
      COMMON/ITDAY/ITDAY
      COMMON/ITDIV/ITDIV
      COMMON/ITFOR/ITFOR,IDFOR
      COMMON/ITHQ/ITHQ
      COMMON/ITIME/ITIME
      COMMON/ITINV/ITINV
      COMMON/ITLEVL/ITLEVL
      COMMON/ITMNTH/ITMNTH
      COMMON/ITQTR/ITQTR
      COMMON/ITWEEK/ITWEEK
      COMMON/ITYEAR/ITYEAR
      COMMON/MBODAY/MBODAY
      COMMON/NBMAX/NBMAX
      COMMON/NENTRY/NENTRY
      COMMON/NFEMAX/NFEMAX
      COMMON/NFIRST/NFIRST
      COMMON/NITEM/NITEM
      COMMON/NLOC/NLOC
      COMMON/NLOCBK/NLOCBK
      COMMON/NTIME/NTIME
     COMMON/ILOCBK/ILOCBK(200)
C
      IF(IDBUG.EQ.1)WRITE(6,113)
  113 FORMAT("*****INITAL.....,INITIALIZE FEL,GASP,RO-FILES,AND EVENTS")
C
C        INITIALIZE THE FUTURE EVENTS LIST
C
      NFEMAX=500
      CALL INFEL
C
C             INITIALIZE GASP FILE SYSTEM
C
```

```
      CALL INSASP(100,27600,4)
C
C
C        SET PARAMETERS FOR BACKORDER FILE
C
      NBMAX=200
      NLOCBK=NBMAX
      DO 10 I=1,NBMAX
   10 ILOCBK(I)=NBMAX+1-I
C
C
C
C       SET TIMING VARIABLES BASED ON 100 TIME UNITS PER DAY
C         7 DAYS/WEEK, 13 WEEKS/QUARTER, 4 QUARTERS/YEAR
      ITDAY=100
      ITWEEK=7*ITDAY
      ITMNTH=4*ITWEEK
      ITQTR=3*ITMNTH
      ITYEAR=4*ITQTR
C
C       SET TIMING VARIABLES FOR MANAGEMENT AND DATA COLLECTION EVENTS
C         CURRENT SIMULATION CLOCK TIME
      ITIME=0
C         CURRENT STATISTICS COLLECTION INTERVAL
      ITINV=1
C         END OF CURRENT QUARTER
      IQTRND=ITQTR
C         TIME      OF FIRST HQ USAF BUDGET AUTHORIZATION
      ITHQ=10
C         TIME      BETWEEN HQ USAF      BUDGET REVISIONS
      IDTHQ=4*ITQTR
C         TIME      OF FIRST DIVISION LEVEL REVIEW
      ITDIV=20
C         TIME      BETWEEN DIVISION LEVEL REVIEWS
      IDDIV=ITMNTH
C         TIME      OF FIRST STOCK LEVEL COMPUTATION
     ITLEVL=30+ITQTR
C         TIME      BETWEEN STOCK LEVEL COMPUTATIONS
     IDLEVL=ITQTR
C        TIME OF FIRST STOCK STATUS REVIEW
      ISTOCK=40
C        TIME BETWEEN STOCK STATUS REVIEWS
      IDSTOC=2*ITWEEK
C         TIME      TO ACTIVATE STATISTICS COLLECTION ROUTINE
      ISTAT=ITWEEK+1
C         TIME      BETWEEN STATISTICAL UPDATES
      IDSTAT=ITWEEK
C        STOP AFTER SIMULATION      INQTR QUARTERS
      ISTOP=INQTR*ITQTR
C
C       PLACE INITIAL MANAGEMENT AND DATA COLLECTION EVENTS ON
C       THE FUTURE EVENTS LIST.
```

```
      C               STAT--REVIEW STOCK STATUS
      C******
      C               LEVELS--COMPUTE INV CONTROL LEVELS
            CALL ENTER(ITLEVL,6,0,0,IDLEVL)
      C
      C         SCHEDULE INITIAL PROVISIONING EVENT
      C
            INPTIM=1
            CALL ENTER(INPTIM,20,0,0,0)
      C               SSTATS--ACCUMULATE STATUS STATISTICS
             CALL ENTER(ISTAT,11,1,0,0)
      C                  SCHEDULE END OF SIMULATION RUN
            CALL ENTER(ISTOP,10,0,0,0)
      C
      C          CREATE FORECAST EVENT
      C
        100 CONTINUE
             ITFOR=ITQTR
             IDFOR=ITQTR
      C               FORUPD--UPDATE DEMAND HISTORY FILES
      *******
      C               DEMPAR--GENERATE DEMAND
      ******
      C
      C               CANCLS--CANCELLATION REVIEW EVENT GOES HERE--
      C
             MBODAY=100
             IDCANB=ITMNTH
      C
      C             SET TRACE EVENTS
      C
             IF(ISTRAC.LE.0) GO TO 200
             CALL ENTER(ITRACE,13,0,0,0)
            CALL ENTER(ISTRAC,13,0,0,0)
       200      CONTINUE
             RETURN
             END
```

```
1       *#RUN=;RIMP/OBJ/INITLB,O,N(BCD,NOGO)
2       *INITLB.S                                                    39
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 ENDOGENOUS EVENTS.
7       C
8             COMMON/IDBUG/IDBUG
9              COMMON/ITRACE/ITRACE,ISTRAC
0              COMMON/IDDIV/IDDIV
1              COMMON/IDLEVL/IDLEVL
2              COMMON/IDSTAT/IDSTAT
3              COMMON/IDTHQ/IDTHQ
4              COMMON/IKDEM/IKDEM
5              COMMON/INQTR/INQTR
6              COMMON/IQTRND/IQTRND
7              COMMON/ISTOCK/ISTOCK,IDSTOC
8              COMMON/ISTOP/ISTOP
9              COMMON/ISTAT/ISTAT
0              COMMON/ITCANB/ITCANB,IDCANB
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/ITLEVL/ITLEVL
8              COMMON/ITMNTH/ITMNTH
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/NFEMAX/NFEMAX
6              COMMON/NFIRST/NFIRST
7              COMMON/NITEM/NITEM
8              COMMON/NLOC/NLOC
9              COMMON/NLOCBK/NLOCBK
0              COMMON/NTIME/NTIME
1             COMMON/ILOCBK/ILOCBK(205)
2       C
3              IF(IDBUG.EQ.1)WRITE(6,113)
4        113 FORMAT(T*****INITAL.....INITIALIZE FEL,GASP,BO-FILES,AND EVENTS")
5       C
6       C        INITIALIZE THE FUTURE EVENTS LIST
7       C
8              NFEMAX=500
9              CALL INFEL
0       C
1       C              INITIALIZE GASP FILE SYSTEM
2       C
```

```
01  11-05-79   19.885   ,S
3             CALL INGASP(100,2,600,4)
4       C                                                              40
5       C
6       C       SET PARAMETERS FOR BACKORDAR FILE
7       C
8             NBMAX=600
9             NLOCBK=NBMAX
0             DO 10 I=1,NBMAX
1          10 ILOCBK(I)=NBMAX+1-I
2       C
3       C
4       C
5       C       SET TIMING VARIABLES BASED ON 100 TIME UNITS PER DAY
6       C         7 DAYS/WEEK, 13 WEEKS/QUARTER, 4 QUARTERS/YEAR
7             ITDAY=100
8             ITWEEK=7*ITDAY
9             ITMNTH=4*ITWEEK
0             ITQTR=3*ITMNTH
1             ITYEAR=4*ITQTR
2       C
3       C       SET TIMING VARIABLES FOR MANAGEMENT AND DATA COLLECTION EVENTS
4       C         CURRENT SIMULATION CLOCK TIME
5             ITIME=0
6       C         CURRENT STATISTICS COLLECTION INTERVAL
7             ITINV=1
8       C         END OF CURRENT QUARTER
9             IQTRND=ITQTR
0       C         TIME      OF FIRST HQ USAF BUDGET AUTHORIZATION
1             ITHQ=10
2       C         TIME      BETWEEN HQ USAF      BUDGET REVISIONS
3             IDTHQ=4*ITQTR
4       C         TIME      OF FIRST DIVISION LEVEL REVIEW
5             ITDIV=20
6       C         TIME      BETWEEN DIVISION LEVEL REVIEWS
7             IDDIV=ITMNTH
8       C         TIME      OF FIRST STOCK LEVEL COMPUTATION
9          ITLEVL=30+ITQTR
0       C         TIME      BETWEEN STOCK LEVEL COMPUTATIONS
1             IDLEVL=ITQTR
4       C       TIME OF FIRST STOCK STATUS REVIEW
3             ISTOCK=40
4       C       TIME BETWEEN STOCK STATUS REVIEWS
5             IDSTOC=2*ITWEEK
6       C         TIME      TO ACTIVATE STATISTICS COLLECTION ROUTINE
7             ISTAT=ITWEEK-1
8       C         TIME      BETWEEN STATISTICAL UPDATES
9             IDSTAT=ITWEEK
0       C       STOP AFTER SIMULATION      INQTR QUARTERS
1             ISTOP=INQTR*ITQTR
2       C
3       C       PLACE INITIAL MANAGEMENT AND DATA COLLECTION EVENTS ON
4       C       THE FUTURE EVENTS LIST.
```

```
C              STAT--REVIEW STOCK STATUS
C*******
C              LEVEL--COMPUTE INV CONTROL LEVELS
               CALL ENTER(ITLEVL,6,0,0,IDLEVL)
C
C         SCHEDULE INITIAL PROVISIONING EVENT
C
          INPTIM=1
          CALL ENTER(INPTIM,20,0,0,0)
C              SSTAT--ACCUMULATE STATUS STATISTICS
           CALL ENTER(ISTAT,11,1,0,0)
C                 SCHEDULE END OF SIMULATION RUN
          CALL ENTER(ISTOP,10,0,0,0)
C
C           CREATE FORECAST EVENT
C
  100 CONTINUE
          ITFOR=ITOTR
          IDFOR=ITOTR
C              FORUPD--UPDATE DEMAND HISTORY FILES
*******
C              DEMPAR--GENERATE DEMAND
******
C
C              CANCLB--CANCELLATION REVIEW EVENT GOES HERE--
C
           MBODAY=100
           IDCANB=ITMNTH
C
C             SET TRACE EVENTS
C
          IF(ISTRAC.LE.0) GO TO 200
           CALL ENTER(ITRACE,18,0,0,0)
          CALL ENTER(ISTRAC,19,0,0,0)
  200       CONTINUE
          RETURN
          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 preprocesing 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.

```
: 01   10-20-79    10.271    ,S
1       *#RUN=;RIME/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,NOUN(2),MSTCD(4),ION,IOR,IPPL,IPPPR
8             COMMON/GSLF/GSLF
9             COMMON/IDBUG/IDBUG
0            COMMON/NBASES/NBASES
1            COMMON/NSRU/NSRU
2             COMMON/ITDAY /ITDAY
3             COMMON/ITMNTH/ITMNTH
4             COMMON/NITEM/NITEM
5              COMMON/NDEM/NDEM
6              COMMON/NDHIS/NDHIS
7              COMMON/INLU/INLU
8              COMMON/INTYPE/INTYPE
9              COMMON/LEBUG/IEBUG
0             COMMON/IBOP/IBOPOR(3),IBOPOR(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/NDEMAC/NDEMAC(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)
)              COMMON/NBOIR/NBOIR(1)
1              COMMON/NBOTR/NBOTR(1)
2             COMMON/REQSIZ/REQSIZ(1)
3             COMMON/REQMAD/REQMAD(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)
)              COMMON/IRETUR/IRETUR(1,24)
1             COMMON/IRL/IRL(1)
2             COMMON/ITL/ITL(1)
```

```
      COMMON/IROL/IROL(1)
      COMMON/IRQTY/IRQTY(1)
      COMMON/RMTBR /RMTBR(1)
      COMMON/RMEAN/RMEAN(1)
      COMMON/RTREND/RTREND(1)
      COMMON/RMAD  /RMAD(1)
      COMMON/RERSUM/RERSUM(1)
      COMMON/KNT   /KNT(1)
      COMMON/GROQ/GROQ(3)
      COMMON/GEOQF/GEOQF(3)
       COMMON/ICDFOR/ICDFOR
C
      CHARACTER BFSN*15
C
C          SET NUMBER OF PERIODS OF DATA INPUT
C
       IDPER=NDEM
C
C
C
      ENTRY INITM1
C
      M=0
      NBP2=NBASES+2
   10 CONTINUE
      READ(7,END=9999)IC1,IC2,IC3,ISEQ,BFSN,COST
       IF(IDBUG.EQ.1) WRITE(6,13)IC1,IC2,IC3,ISEQ,BFSN,COST
   13 FORMAT(3I5,2X,"ISEQ=",I8,"  FSN=",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
      M=(N-1)*NBP2 + NN
      LTADM(M)=2
      LTPROD(M)=8
      IBST(M)=10
      IDST(M)=20
      IDEST(M)=30
      UCOST(M)=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("******INITEM.  NITEM=",I5,"     NSRU=",I5)
```

```
      RETURN
C
C----------BEGIN ITEM INITIALIZATION LOOP-----
C
      ENTRY INITM2
        IF(IDBUG.EQ.1)WRITE(6,123)
  123 FORMAT("BEGIN INITM2.....ITEM INITIALIZATION")
C
      DO 100 NN=1,NITEM
      N=NN
C             INDICATE ITEM HAS NDHIS PERIODS OF DEMAND HISTORY
C
       NDENT(N)=NDHIS
C
C     ZERO DEMAND HISTORY RECORDS
        NRETAC(N)=0
       NDEMAC(N)=0
        NREQAC(N)=0
C       SET INVENTORY DUE-IN TO ZERO
   70 INVDUE(N)=0
        NORDPT(N)=0
      INWIP(N)=0
C
C       SET INITIAL BACKORDER COUNTERS TO ZERO
C
       NBOTU(N)=0
        NBOIU(N)=0
        NBOIR(N)=0
        NBOTR(N)=0
       NBOPT(N)=0
C
C           ESTABLISH BEGINNING INVENTORY LEVELS
C
      INVACT(N)=0
  100       CONTINUE
C
       RETURN
C
C           IF END OF FILE IS READ, PRINT MESSAGE, SET NITEM=0,
C                   AND THEN RETURN
 9999 CONTINUE
      NITEM=0
        NSRU=0
        WRITE(6,9993)NITEM,NSRU
 9993 FORMAT(/////"READ END OF FILE OF THE FOLLOWING VALUES WERE SET",
     &       //T20,"NITEM=",I8,"     NSRU=",I8////)
       RETURN
      END
```

T 01 10-20-79 16.696 CS

```
*@RUN=LRINE/ODE/INITM2.O(BCD,NOGOY
*INITM2.S
      SUBROUTINE INITEM
C         READ IN AND INITIALIZE DATA FOR A NEW LRU/SRU SET
C
        CHARACTER ALC,FSN,UM,NOUN,NGPCD
        COMMON/FSN/ALC,FSN(4),UNTNOUN(2),NGPCD(4),ION,IOK,IPPR,IPRPS
       COMMON/CCLY/CCLY
      COMMON/NBUG/NBUG
      COMMON/NBASES/NBASES
      COMMON/NSRU/NSRU
      COMMON/IDENT/IDENT
      COMMON/NLAN/NLAN,MLAN
      COMMON/NREPL/NREPL,MREPL
      COMMON/NGROUP/NGROUP
      COMMON/NPERP/NPERP
      COMMON/NLORP/NLORP
      COMMON/NQTB/NQTB
C-------------------------------
       COMMON/IDAY /IDAY
       COMMON/IMNTH/IMNTH
       COMMON/NITEM/NITEM
        COMMON/NDEM/NDEM
        COMMON/NDNIS/NDNIS
        COMMON/INTYPE/INTYPE
       COMMON/IDOP/IDOPON(3),IDOPON(8)
       COMMON/IBVACT/IBVACT(1)
      COMMON/IART/IART(1)
      COMMON/IDRT/IDRT(1)
      COMMON/IDOET/IDOET(1)
       COMMON/NDBNAC/NDBNAC(1)
        COMMON/NRETAC/NRETAC(1)
        COMMON/NRBOAC/NRBOAC(1)
        COMMON/NDEMND/NDEMND(1,28)
        COMMON/NRETUR/NRETUR(1,28)
        COMMON/NRBO/NRBO(1,24)
       COMMON/NDENT/NDENT(1)
       COMMON/INVDUE/INVDUE(1)
      COMMON/NWIP/NWIP(1)
       COMMON/NBOPT/NBOPT(1)
       COMMON/NBOTU/NBOTU(1)
        COMMON/NBOIU/NBOIU(1)
        COMMON/NBOIR/NBOIR(1)
        COMMON/NBOTR/NBOTR(1)
       COMMON/LTPROD/LTPROD(1)
       COMMON/LTADM/LTADM(1)
       COMMON/UCOST/UCOST(1)
       COMMON/ISUL/ISUL(1)
       COMMON/IRL/IRL(1)
       COMMON/ITL/ITL(1)
       COMMON/IROL/IROL(1)
       COMMON/IROTY/IROTY(1)
```

INITM2

```
C
      CHARACTER [illegible]
      DIMENSION [illegible],JSHIP(8)
C
      ENTRY [illegible]
C      SET RECORD [illegible] OF [illegible] FILE TO 10 [illegible].
C
      CALL [illegible](10,13)
C
C
C
C
      N=0
      ND[illegible]=[illegible]
   10 CONTINUE
      READ(7,[illegible])IC1,IC2,IC3,[illegible],BPSU,COST
     &,JOPX,[illegible],JDRT,JBRT,JDORT,JTADM,JTPROD,JBDORD,
     &       JOVSHP,JSHIP
      IF([illegible]) WRITE(6,13)IC1,IC2,IC3,[illegible],BPSU,COST
     &,JOPX,[illegible],JDRT,JBRT,JDORT,JTADM,JTPROD,JBDORD
     &,JOVSHP,JSHIP
   13 FORMAT([illegible],2X,"ISEQ=",I8," PN=",A10," UCOST=",F10.2
     &/T20,"    JOPX  JBDVT    JDRT    JBRT  JDORT  JTADM JTPROD JBDORD"
     &," JOVSHP  JSHIP1,2,...,8"
     &,/T20,9I7,8X,8I3)
C
      IF(IC1.EQ.0) GO TO 40
      N=N+1
C
C
C      SET DEPOT PARAMETERS.
C
      N=(N-1)*[illegible]
      ND=N+1
      LTADM(ND)=JTADM
      LTPROD(ND)=JTPROD
      IBRT(ND)=JBRT
      IDRT(ND)=JDRT
      IDORT(ND)=JDORT
      UCOST(ND)=COST
C
C      SET BASE PARAMETERS
C
      DO 50 KB=1,NBASES
      NB=KB+N+1
      LTADM(NB)=JBDORD
      LTPROD(NB)=JSHIP(KB)
      IBRT(NB)=JBRT
      IDRT(NB)=JBDVT+IDRT
      IDORT(NB)=JBDVT+JDORT
      UCOST(NB)=COST
   50 CONTINUE
```

```
      C
      C        SET OVERHAUL FACILITY PARAMETERS
      C
             MD=M+NBASES+1
             LTADR(MD)=JBDORD
             LTPROD(MD)=JOVSHP
             IBRT(MD)=JDCST
             IDRT(MD)=JBDVT+JBRT
             IDORT(MD)=JDORT
             VCOST(MD)=COST
             GO TO 10
      C
      C          SET NITEM AND NSRU
      C
          40 CONTINUE
             NITEM=(NBASES+2)*N
             NSRU=0
             IF(N.GT.1)NSRU=N-1
             IF(IMENU.EQ.1)WRITE(6,113)NITEM,NSRU
         113 FORMAT(" ***INITEM.  NITEM=",I5,"     NSRU=",I5)
             INGRP=NSRU+1
      C
      C      READ LEVELS FOR QTRS 1,3,5,7,9 AND ALL GROUPS
      C                FOR THIS LAMBDA
      C            AND WRITE TO WORK FILE 19.
      C
             IF(IMENU.EQ.1)WRITE(6,43)
             DO 60 KQ=1,NQTR,2
             DO 55 IT=1,INGRP
             READ(9,33,END=70)IDENT,KLAM,LGRP,KQTR,NUMB,IOVSL,
            &KD,KC,(IBSL(K),K=1,NBASES)
          33 FORMAT(T10,5I5,T31,I3,T48,10I3)
             IF(IMENU.EQ.1)WRITE(6,53)
            &     IDENT,KLAM,LGRP,KQTR,NUMB,IOVSL,
            &KD,KC,(IBSL(K),K=1,NBASES)
      C
          43 FORMAT(" ***INITEM1 INPUT LEVELS--  IDENT  KLAM LGRP  KQTR  NUMB",
            &  " IOVSL    KD    KC   IBSL+1,2,...")
          53 FORMAT( 26X,8I6,8I3)
      C
      C            CHECK THAT INDEXES ARE AS EXPECTED
      C
             IF(KLAM.NE.MLAM) GO TO 399
             IF(LGRP.NE.NGROUP) GOTO 399
             IF(KQTR.NE.KQ) GO TO 399
      C
      C            DATA CHECKS.  WRITE ENTRIES TO FILE 11
      C       OUTPUT DUPLICATE LEVELS FOR QTRS KQ AND KQ+1
      C
             INDEX=46*(KQ-1)+IT
             WRITE(11'INDEX)IOVSL,KD,KC,IBSL
             INDEX=46*KQ + IT
```

```
          WRITE(11'INDEX)IOVSL,KD,KC,IBSL
      55 CONTINUE
      60 CONTINUE
C
     70   CONTINUE
          RETURN
C
C             LEVEL RECORD PARAMETER WAS NOT AS EXPECTED.
C                STOP RUN.
     399 CONTINUE
          WRITE(6,68)"*****INITM1--LEVELS PARAMETERS DON'T MATCH",
     &            "  MLAM=",MLAM,"  NGROUP=",NGROUP,"  KO=",KO
          WRITE(6,63 )"-----LEVELS READ ARE..."
          WRITE(6,58)
     &       IDENT,KLAM,LEUGP,KOTR,NUMBETOVSL,
          &KD,KC,(IBSL(N),N=1,NBASES)
      63 FORMAT(V)
          WRITE(6,62)"----INITM2--STOP RUN."
          STOP
C
C------------------------BEGIN ITEM INITIALIZATION LOOP-------
C
   ENTRY INITM2
            IF(IDBUG.EQ.1)WRITE(6,123)
      123 FORMAT("BEGIN INITM2...,ITEM INITIALIZATION")
C
C
          DO 100 NN=1,NITEM
          N=NN
C            SET INVENTORY DUE-IN TO ZERO
           INVDUE(N)=0
          INWIP(N)=0
C
C           SET INITIAL BACKORDER COUNTERS TO ZERO
C
           NBOTQ(N)=0
            NBOTU(N)=0
            NBOZR(N)=0
            NBOTR(N)=0
           NBOPT(N)=0
C
C             ESTABLISH BEGINNING INVENTORY LEVELS
C
          INVACT(N)=0
    100         CONTINUE
C
          IF(IDBUG.EQ.1)WRITE(6,63)"----EXIT INITM2"
           RETURN
C
C               IF END OF FILE IS READ PRINT MESSAGE, SET NITEM=0,
C                      AND THEN RETURN
    9999 CONTINUE
```

```
           NITEM=0
             NSRU=0
             WRITE(6,9993)NITEM,NSRU
     9993 FORMAT(/////"READ END OF FILE 07,THE FOLLOWING VALUES WERE SET",
     &          //T20,"NITEM=",I8,"      NSRU=",I8//1/)
           RETURN
          END
**W     7 MEMORY EXPANDED, USE $LIMITS OR CORE= 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   .S

```
*#RUN=IRIMS/SB6/ITRSL2.O(BCD,NOGO)
*ITRSL2.S
      SUBROUTINE ITRSLT
      COMMON/C2GNT/ITWRT,IOUT,IGRAPH,ISUMRY
       COMMON/IDENT/IDENT
       COMMON/NLAM/NLAM,MLAM
       COMMON/NREPL/NREPL,MREPL
       COMMON/NGROUP/NGROUP
      COMMON/NFGRP/NFGRP
      COMMON/INQTR/INQTR
       COMMON/ITINV/ITINV
       COMMON/IORDER/IORDER(16,3,6)
       COMMON/IBAKDI/IBAKDI(16,3,6)
       COMMON/IFILLI/IFILLI(16,3,6)
      COMMON/IFILLT/IFILLT(16,3,6)
       COMMON/IREQI/IREQI(16,3,6)
      COMMON/IREQT/IREQT(16,3,6)
      COMMON/IBODAI/IBODAI(16,3,6)
      COMMON/IBODAT/IBODAT(16,3,6)
C
      DIMENSION IMEAS(16,10),JMEAS(16,10)
      DIMENSION ITOTL(10)
      DATA NMEAS/10/
C
C            IMEAS(I,J) = VALUE OF MEASURE J FOR QTR I OF THIS
C                         REPLICATION.
C            JMEAS(I,J) = CUMULATIVE VALUE OF MEASURE J FOR QTR I
C                         FOR ALL REPLICATIONS UP TO AND INCLUDING
C                         THIS ONE.
C            NMEAS = 10 = NO. OF MEASURES RECORDED BY THIS ROUTINE.
C
C            ON FIRST REPLICATION, ZERO THE JMEAS-ARRAY
C              WHEN THE FIRST ITEM IS PROCESSED.
C
      IF(NGROUP.NE.NFGRP) GO TO 20
      IF(MREPL.NE.1) GO TO 20
      DO 10 I=1,16
      DO 10 J=1,NMEAS
       JMEAS(I,J)=0
   10 CONTINUE
   20 CONTINUE
C
C            COMPUTE STATISTICS FOR THIS REPLICATION
C
      DO 50 I=1,INQTR
        IMEAS(I,1)=IORDER(I,3,3)-JMEAS(I,1)
        IMEAS(I,2)=IORDER(I,3,4)-JMEAS(I,2)
       IMEAS(I,3)=IBODAT(I,2,3)/100 -JMEAS(I,3)
      IMEAS(I,4)=IBODAT(I,2,4)/100 -JMEAS(I,4)
      IMEAS(I,5)=IFILLT(I,2,1)-JMEAS(I,5)
        IMEAS(I,6)=IFILLT(I,2,2)-JMEAS(I,6)
      IMEAS(I,7)=IREQT(I,2,1)-JMEAS(I,7)
```

```
          IMEAS(I,8)=IBDOT(I,2,2)-JMEAS(I,8)
      C
          IMEAS(I,9)=IBODAT(I,2,1)/100 -JMEAS(I,9)
          IMEAS(I,10)=IBODAT(I,2,2)/100 -JMEAS(I,10)
      C
      C
   50 CONTINUE
      C
      C
      C        RECORD CURRENT VALUES FOR USE ON THE NEXT REPLICATION
      C
          DO 300 I=1,INQTR
          DO 290 J=1,NMEAS
           JMEAS(I,J)=JMEAS(I,J)+IMEAS(I,J)
  290 CONTINUE
  300 CONTINUE
      C
      C
      C        OUTPUT RESULTS FOR THIS REPLICATION
      C
      C
          WRITE(6,1)NGROUP,IDENT,MLAM,MREPL
    1     FORMAT(1H1," *****ITRSLT...     NGROUP=",1X,I2,"  IDENT= ",I2,
      &     ,"  MLAM=",I2,"   MREPL =",I2)
          WRITE(6,2)
      C
          IF((MLAM.GT.1).OR.(MREPL.GT.1))GO TO 180
          KNT=8
          IF(INQTR.LT.KNT)KNT=INQTR
          WRITE(15,3)KNT,NGROUP,IDENT
          IF(INQTR.GT.8)WRITE(16,3)INQTR,NGROUP,IDENT
    3     FORMAT(1H1///T30,I2,"-QTR TOTALS"/T25,"NGROUP =",I3,
      &     "   IDENT =",I3)
          WRITE(15,2)
          IF(INQTR.GT.8)WRITE(16,2)
    2 FORMAT(//T20,"DEBOT",T40,"DEBOH",T60,3("BASE",16X)/
      &     T17,"IORDER=S",T39,"IBODAT",T58,"IFILLT",T79,"IBBOT",
      &     T99,"IBODAT"/
      &     T6,"I",T15,5("LRU        SRU        ")/)
      C
      C
      C
  180 CONTINUE
          DO 200 I=1,INQTR
           IF(ITHRY.EQ.2) PUNCH 5,NGROUP,IDENT,MLAM,MREPL,I,
      &     (IMEAS(I,J),J=1,NMEAS)
    5 FORMAT(5I2,2I10,8I6,"II")
          WRITE(6,193)I,(IMEAS(I,J),J=1,NMEAS)
  193 FORMAT(1H ,I5,10X10)
  200 CONTINUE
      C          COMPUTE 8 AND 16 QTR TOTALS
      C
```

```
5             INDEX=0
6             IF(INQTR.GT.INDEX)INDEX=INQTR
7             DO 320 J=1,NMEAS
8              ITOTL(J)=0
9              DO 310 I=1,INDEX
                ITOTL(J)=ITOTL(J)+IMEAS(I,J)
1             310 CONTINUE
2            320 CONTINUE
3       C
4       C           IF ITWRT=1, PUNCH 8-QTR SUMMARIES
5       C
6             IF(ITWRT.EQ.1) PUNCH 15,NGROUP,IDENT,MLAN,MREPL,INDEX,
7       &        (ITOTL(J),J=1,NMEAS)
8          15 FORMAT(5I2,2I10,8X6,"AA")
9             WRITE(6,333)INDEX,(ITOTL(J),J=1,NMEAS)
         333 FORMAT(//1X,I2,"QTR TOTALS"/6X,10I10)
1       C
2             IF(MREPL.EQ.1)WRITE(15,343)
3          343  FORMAT(/)
4             WRITE(15,353)MLAN,MREPL,(ITOTL(J),J=1,NMEAS)
5          353  FORMAT(2I3,10I10)
6       C
7             IF(INQTR.EQ.8) GO TO 400
8              DO 360 J=1,NMEAS
9             DO 350 I=9,INQTR
               ITOTL(J)=ITOTL(J)+IMEAS(I,J)
1          350 CONTINUE
2          360 CONTINUE
3       C
4       C           IF ITWRT=1,PUNCH 16-QTR SUMMARIES
5       C
6             IF(ITWRT.EQ.1) PUNCH 35,NGROUP,IDENT,MLAN,MREPL,INQTR,
7       &           (ITOTL(J),J=1,NMEAS)
8          35 FORMAT(5I2,2I10,8X6,"MM")
9             WRITE(6,383)INQTR,(ITOTL(J),J=1,NMEAS)
0       C
1             IF(MREPL.EQ.1)WRITE(16,343)
2             WRITE(16,853)MLAN,MREPL,(ITOTL(J),J=1,NMEAS)
3       C
4          400 CONTINUE
5             RETURN
6              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.

r 01 10-20-79 10.330 SKU.S

```
1       ******KNSKU.S
2             FUNCTION KNSKU(N)
3       C
4       C       THIS FUNCTION COMPUTES THE STATISTICS INDEX K
5       C            ASSOCIATED WITH NKSKU=N.
6             COMMON/NBASES/NBASES
7        NSTLOC=NBASES+2
8        NDEX=MOD(N,NSTLOC)
9           IF(N.LE.NSTLOC) GO TO 10
10      C
11      C            N IS AN SRU
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.

```
      *RUN*,TIME/OBJ/LEVEL,O(BCD,NOGO)
      *LEVEL,S   COMPUTE CONTROL LEVELS
            SUBROUTINE LEVEL(NN)
      C
      C          THIS ROUTINE COMPUTES      REORDER,STOCK OBJECTIVE,RETENTION,
      C           TERMINATION, AND SUPPORT LEVELS.
      C
             COMMON/IHBUG/IHBUG
             COMMON/COSHRT/COSHRT
            COMMON/COSHLD/COSHLD
            COMMON/COSORD/COSORD(3)
             COMMON/CSTBRK/CSTBRK
            COMMON/GSULP/GSULP
            COMMON/GRLP/GRLP
            COMMON/GSLP/GSLP
            COMMON/GTLP/GTLP
            COMMON/ITLEVL/ITLEVL
            COMMON/IDLEVL/IDLEVL
            COMMON/NITEM/NITEM
           COMMON/NBASES/NBASES
            COMMON/POLICY/ICDEOQ,ICDSL,EOQMAX,EOQMIN,SLMAX,SLMIN,RLP,TLP,SULP
             COMMON/ICDSLL/ICDSLL
            COMMON/ADR/ADR(1)
            COMMON/LTADM/LTADM(1)
            COMMON/LTPROD/LTPROD(1)
            COMMON/IRQTY/IRQTY(1)
            COMMON/IRL/IRL(1)
            COMMON/IROL/IROL(1)
            COMMON/ISUL/ISUL(1)
            COMMON/ITL/ITL(1)
            COMMON/RSIGLT/RSIGLT(1)
             COMMON/REQSIZ/REQSIZ(1)
           COMMON/UCOST/UCOST(1)
            DATA Z/1.0/
            DATA ITLMIN,IRLMIN/99999999,99999999/
      C
      C                COMPUTE LEVELS FOR ITEM NN.  IF NN=0,
      C                     COMPUTE LEVELS FOR ALL ITEMS,
      C                     THAT IS, ITEMS 1,2,....NITEM
      C
            NLOC=NBASES+2
            NP=NN
            NL=NN
            IF(NN.EQ.0)NP=1
            IF(NN.EQ.0)NL=NITEM
      C
            DO 3000 NNN=NP,NL
             N=NNN
      C
      C
      C          COMPUTE PLANNING FACTORS
      C
```

```
      AMDA=COSHRT
      RLTM=LTADM(N)+LTPROD(N)
      IF(RLTM.LT.1.)RLTM=0.5
****
C                FOR NOW, SET ANNUAL DEMAND=12
C              FOR THE OVERHAUL FACILITY, SET AD=24.
      AD=12.
      KMOD=MOD(N,NIOC)
      IF(KMOD.EQ.0) AD=24.
C
C           NOTE....LEADTIME IS IN DAYS
C
      RLT=AD*RLTM/365.
      RMR=AD/12.
      UC=UCOST(N)
      ADDR=UC*AD
C
C
C          FOR NOW, SET Q=1
C
      Q=1
C
C                    FOR NOW, SET SL=0
      SL=0
C
C
C          CHECK       ORDER SIZE LIMITS
  200 CONTINUE
      EMX=EOQMAX*RMR
      IF(Q.GT.EMX) Q=EMX
      EMX=EOQMIN*RMR
      IF(Q.LT.EMX) Q=EMX
      IF(Q.LT.      1.)Q= 1.
C
C
C           LIMITS BASED ON MONTHS OF SUPPLY
C
  520       CONTINUE
C           LOWER LIMIT
      SLM=SLMIN*RMR
      IF(SL.LT.SLM)SL=SLM
C           UPPER LIMIT
      SLM=SLMAX*RMR
      IF(SL.GT.SLM)SL=SLM
C
C
      GO TO 600
C*************************************************
C
  600       CONTINUE
C
C     COMPUTE LEVELS
```

```
! 01  10-20-79   10.539   S   COMPUTE CONTROL LEVELS

15      C
16            IRQTY(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))+GRLF*RMR+0.5
10            IF(ITL(N).LE.ITLMIN) ITL(N)=ITLMIN
 1            IF(IRL(N).LE.IRLMIN)IRL(N)=IRLMIN
12            ISUL(N)=GSULF*RMR
13      C
14      C
15          IF(IHBUG.NE.1) GO TO 2990
16            WRITE(6,8903)N,IRQTY(N),IROL(N),ITL(N),IRL(N),ISUL(N)
17       8903       FORMAT(4X,'****LEVELN--N=',I5,'  IRQTY=',I5,'  IROL=',I5,
18      &           '  ITL=',I5,'  IRL=',I5,'  ISUL=',I5)
19       2990 CONTINUE
10       3000 CONTINUE
 1            RETURN
12            END
```

```
*#RUN=IRIME/6BS/LEVEL2.O(BCD,NOGO)
*LEVEL2,S
                SUBROUTINE LEVEL(NN)
      C          THIS ROUTINE COMPUTES LEVELS FOR ALL
      C      (NSRU+1) STOCK KEEPING UNITS BY READING THE NEXT (NSRU+1)
      C     SETS OF LEVELS RECORDS ON FILE 11.  IT THEN COMPUTES
      C       THE STOCK KEEPING UNIT NUMBER ASSOCIATED WITH EACH
      C       NSN, AND SETS THE APPROPRIATE COMMON VARIABLES.
      C
            COMMON/IHBUG/IHBUG
             COMMON/ITLEVL/ITLEVL
             COMMON/IDLEVL/IDLEVL
            COMMON/NSRU/NSRU
             COMMON/NITEM/NITEM
            COMMON/NBASES/NBASES
             COMMON/LTADM/LTADM(1)
             COMMON/LTPROD/LTPROD(1)
             COMMON/IROTY/IROTY(1)
             COMMON/IBL/IBL(1)
             COMMON/IROL/IROL(1)
             COMMON/ISUL/ISUL(1)
             COMMON/ITL/ITL(1)
            COMMON/ITINV/ITINV
      C
      C
            DIMENSION IBSL(8)
      C
            KQ=ITINV
            ITGRP=NSRU+1
            NLOC=NBASES+2
            DO 300 I=1,ITGRP
      C
      C          INPUT LEVELS
            INDEX=16*(KQ-1)+I
            READ(11'INDEX)IOVSL,KD,KC,IBSL
            IF(IHBUG.EQ.1)WRITE(6,33)KQ,I,IOVSL,KD,KC,IBSL
         33 FORMAT(" ****LEVEL2--QTR=",I3,"  I=",I3,"  IOVSL=",I3,
      &            "  KD=",I3,"  KC=",I3,"  IBSL=",8I3)
      C
      C
      C          COMPUTE DEPOT AND OVERHAUL SKU NUMBERS
      C
            NSKUDP=(I-1)*NLOC+1
            NSKUOV=NSKUDP+NLOC-1
      C
      C          SET REORDER LEVELS FOR DEPOT AND OVERHAUL FACILITIES
      C
            IROL(NSKUDP)=KD+KC
            IROL(NSKUOV)=IOVSL
      C
      C          SET BASE LEVELS
      C
```

```
      DO 50 NB=1,NBASES
       NSKU=NSKUDP+NB
      IROL(NSKU)=IBSL(NB)
   50 CONTINUE
C
C           SET OTHER LEVELS TO CONSTANT VALUES
C
      DO 80 N=NSKUDS,NSKUOV
       IROTY(N)=1
       ITL(N)=9999999
       IRL(N)=9999999
       ISUL(N)=0
   80 CONTINUE
C
C
      IF(INBUG.NE.1) GO TO 2990
      DO 90 N=NSKUDS,NSKUOV
        WRITE(6,8903)N,IROTY(N),IRBL(N),ITL(N),IRL(N),ISUL(N)
 8903       FORMAT(4X,'****LEVELZ0-N=',I5,'  IROTY=',I5,'  IROL=',I5,
     &    '  ITL=',I5,'  IRL=',I5,'  ISUL=',I5)
   90 CONTINUE
 2990 CONTINUE
 3000 CONTINUE
       RETURN
       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(16,3,6)
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
```

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 immediatedly 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.

F 01 10-20-79 10.536 S

```
1       *$RUN*;$IME/OBJ/ORDER.O(BCD,NOGO)
2       *ORDER.S
3             SUBROUTINE ORDER(N,IBQ,JTIME)
4       C        THIS ROUTINE UPDATES STATISTICS TO REFLECT AN ORDER FOR IBQ
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/ISMORD/ISMORD(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/IBOPSM(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
```

```
            JTIME=ITIME + ITDAY*(LTADM(N ))
            GO TO 30
      C
          ENTRY ORDERV(N,IBO,JTIME)
      C                ORDER DIRECT FROM VENDOR IBO UNITS OF ITEM N,
      C                     FOR DELIVERY AT TIME=JTIME
      C
      C              COMPUTE SKU OF DEPOT = NDEP
      C
            NBP2= NBASES + 2
            NS=(N-1)/NBP2
            NDEP=NS*NBP2 + 1
      C
      C              SET FLAGS FOR INITIAL PROVISIONING
      C
         10 CONTINUE
            IPHASE=1
            JTYPE=2
            IP3=N
            IP4=IBO
            IP5=0
            GO TO 30
      C
      C                LOCATION IS A DEPOT
      C
         20 CONTINUE
            JTIME=ITIME + ITDAY*(LTADM(N)+LTPROD(N))
            JTYPE=2
            IP3=N
            IP5=0
         30 CONTINUE
      C
      C
      C                PLACE ORDER FOR IBO UNITS OF ITEM IP3
      C
            CALL ENTER(JTIME,JTYPE,IP3,IBO,IP5)
      C           UPDATE          STATUS STATISTICS
             IOBLIG=FLOAT(IOBLIG)+FLOAT(IBO)*UCOST(N)
             INVDUE(N)=INVDUE(N)+IBO
              DVORD=FLOAT(IBO)*UCOST(N)
              IF(ITIME .LE. 0) GO TO 100
             IF(IPHASE.EQ.1)CALL CUM(IORDER,IBO,NDEP)
             IF(IPHASE.EQ.2)CALL CUM(IORDER,IBO,N)
             DVORD=FLOAT(IBO)*UCOST(N)
             IF(DVORD        .GE. CSTBRK) CALL CUM(ILGORD,IBO,N)
             IF(DVORD        .LT. CSTBRK) CALL CUM(ISMORD,IBO,N)
              GO TO 200
        100        IDVORD=IFIX(DVORD)
              IF(DVORD .GE. CSTBRK) GO TO 150
              IBOPSM(1)=IBOPSM(1) + 1
              IBOPSM(2)=IBOPSM(2) + IBO
              IBOPSM(3)=IBOPSM(3) + IDVORD
```

```
05              GO TO 200
06        150       IBOPLG(1)=IBOPLG(1) +1
07              IBOPLG(2)=IBOPLG(2) + IBQ
08              IBOPLG(3)=IBOPLG(3) + IDVORD
09        200       CONTINUE
10            IF(IDBUG.NE.1) GO TO 22
11              BUYDOL = UCOST(N)*IBQ
12              WRITE(6,8000)N,IBQ,INVDUE(N),NLOC,UCOST(N),BUYDOL
13        8000 FORMAT("******ORDER=-N=",I3,"  IBQ=",I5,"  INVDUE=",I5,
14        8"  NLOC=",I5,"  UCOST=",F10.2,T110,"  BUY$="
15            &,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.

```
      *$RUN=;RIME/OBJ/OUTREP.O(BCD.NOG6)
      *OUTREP,S  RIME SHORT-FORM SUMMARY REPORT
            SUBROUTINE OUTREP(KF,KL)
             COMMON/IDENT/IDENT
             COMMON/MLAM/MLAM,MLAM
             COMMON/MREPL/MRZPL,MREPL
             COMMON/NGROUP/NGROUP
            COMMON/NFGRP/NFGRP
            COMMON/IORDER/ IORDER(16,3,6)
           COMMON/IWFP/ IWFP(16,3,6)
            COMMON/IBAKDT/IBAKDT(16,3,6)
            COMMON/IBODAI/IBODAI(16,3,6)
            COMMON/IBODAT/IBODAT(16,3,6)
            COMMON/IREQT/IREQT(16,3,6)
            COMMON/IFILLT/IFILLT(16,3,6)
            COMMON/INQTR/ INQTR
             DIMENSION FIL(6)
            DIMENSION ITORD(6),ITWFP(6),ITBAKD(6),ITREQ(6),ITFILL(6)
            DIMENSION TYPE(3,3)
            DATA TYPE/"ACTI","ONS/","FSN ","UNIT","S   ","    ","DOLL","ARS ",
     &          "    "/
     C
     C                  PRODUCE REPORT FOR MEASURES J=1,3
     C                           WHERE J=1   ACTIONS/FSN
     C                                 J=2   UNITS
     C                                 J=3   DOLLARS
     C
            DO 1000 J=KF,ML
     C
     C
     C                  PRINT HEADINGS.
     C
            WRITE(6,60)(TYPE(K,J),K=1,3),J
          60 FORMAT("1"///T20,"SIMULATION RESULTS IN ",3A4,2X,"(J=",I1," )")
            WRITE(6,63)NGROUP,IDENT,MLAM,MREPL,INQTR
          63  FORMAT(/T10,"NGROUP =",I3,"   IDENT =",I3,
     &          "   MLAM =",I3,"   MREPL =",I3,"   INQTR =",I3,
     &           )
     C
     C                  INITIALIZE ARRAYS AND VARIABLES.
     C
            DO 20 K=1,6
            ITORD(K)=0
            ITWFP(K)=0
            ITBAKD(K)=0
            ITREQ(K)=0
            ITFILL(K)=0
          20 CONTINUE
            IGDORD=0
            IGDWFP=0
            IGDBAK=0
            IGDREQ=0
```

```
      IGDFIL=0
C
C                  COMPUTE SUMMATIONS FOR INQTR QUARTERS FOR EACH OF THE
C                 AGGREGATE CATEGORIES(K=1,6).
C                  WHERE K=1 REPRESENTS LRU"S AT BASE
C                        K=2 REPRESENTS SRU"S AT BASE
C                        K=3 REPRESENTS LRU"S AT DEPOT
C                        K=4 REPRESENTS SRU"S AT DEPOT
C                        K=5 REPRESENTS LRU"S AT OVERHAUL FACILITY
C                        K=6 REPRESENTS SRU"S AT OVERHAUL FACILITY
C
      DO 100 K=1,6
      DO  50 I=1,INQTR
      ITORD(K)=ITORD(K)+IORDER(I,J,K)
      ITWFP(K)=ITWFP(K)+IWFP(I,J,K)
      ITBAKD(K)=ITBAKD(K)+IBAKDT(I,J,K)
      ITREQ(K)=ITREQ(K)+IREQT(I,J,K)
      ITFILL(K)=ITFILL(K)+IFILLT(I,J,K)
   50 CONTINUE
C
C                  COMPUTE GRAND TOTALS.
      IGDORD=IGDORD+ITORD(K)
      IGDWFP=IGDWFP+ITWFP(K)
      IGDBAK=IGDBAK+ITBAKD(K)
      IGDREQ=IGDREQ+ITREQ(K)
      IGDFIL=IGDFIL+ITFILL(K)
  100 CONTINUE
C
C                  COMPUTE INTERMEDIATE TOTALS FOR "ORDERS PLACED."
C
      INTOD1=ITORD(1)+ITORD(2)
      INTOD2=ITORD(3)+ITORD(4)
      INTOD3=ITORD(5)+ITORD(6)
C
C                  COMPLETE INTERMEDIATE TOTALS FOR "WAIT FOR PARTS".
C
      INTWP1=ITWFP(1)+ITWFP(2)
      INTWP2=ITWFP(3)+ITWFP(4)
      INTWP3=ITWFP(5)+ITWFP(6)
C
C                  COMPUTE INTERMEDIATE TOTALS FOR "BACKORDER WEEKS".
C
      INTBK1=ITBAKD(1)+ITBAKD(2)
      INTBK2=ITBAKD(3)+ITBAKD(4)
       INTBK3=ITBAKD(5)+ITBAKD(6)
C
C                  COMPUTE INTERMEDIATE TOTALS FOR "TOTAL REQUISITIONS".
C
      INTRQ1=ITREQ(1)+ITREQ(2)
      INTRQ2=ITREQ(3)+ITREQ(4)
      INTRQ3=ITREQ(5)+ITREQ(6)
C
```

```
      C                COMPUTE INTERMEDIATE TOTALS FOR "TOTAL FILLS".
      C
            INTFL1=ITFILL(1)+ITFILL(2)
            INTFL2=ITFILL(3)+ITFILL(4)
            INTFL3=ITFILL(5)+ITFILL(6)
      C
      C                  COMPUTE "FILL PERCENTAGES".
      C
           DO 200 K=1,6
            FIL(K)=FILLF(ITFILL(K),ITREQ(K))
          200 CONTINUE
            FILP1=FILLF(INTFL1,INTRQ1)
            FILP2=FILLF(INTFL2,INTRQ2)
            FILP3=FILLF(INTFL3,INTRQ3)
            FILGD=FILLF(IGDFIL,IGDREQ)
      C
      C
      C                  PRINT INTERMEDIATE HEADINGS
      C
            WRITE (6,210)(I,I=1,6)
          210 FORMAT(///,17X,6("(",I1,")",7X)/
     &           /,11X,"REQUISITIONS",4X,"WAIT"/
     &             " ",16X,"TO        FOR   BACKORDER   TOTAL     TOTAL   ",
     &         "    FILL"/,
     &           " ",13X,"SUPPLIERS   PARTS    WEEKS  REQUISITIONS FILLS",
     &         "   PERCENTAGE"/
     &             " ",13X,6("=========",2X))
      C
      C                  WRITE OUT RESULTS.
      C
           WRITE(6,250) ITORD(1),ITWFP(1),ITBAKD(1),ITREQ(1),
     &         ITFILL(1),FIL(1)
          250 FORMAT(//,"BASE LRU "5I10,F10.3)

            WRITE(6,260) ITORD(2),ITWFP(2),ITBAKD(2),ITREQ(2),ITFILL(2),FIL(2)
          260 FORMAT(//,"BASE SRU ",5I10,F10.3)
            WRITE(6,270) INTOD1,INTWP1,INTBK1,INTRQ1,INTFL1,FILP1
          270 FORMAT(" ",13X,6("=========",2X)/
     &             /"    TOTAL",5I10,F10.3)
           WRITE(6,280) ITORD(3),ITWFP(3),ITBAKD(3),ITREQ(3),ITFILL(3),FIL(3)
          280 FORMAT(///,"DEPOT LRU",5I10,F10.3)
            WRITE(6,290) ITORD(4),ITWFP(4),ITBAKD(4),ITREQ(4),ITFILL(4),FIL(4)
          290 FORMAT(//,"DEPOT SRU",5I10,F10.3)
           WRITE(6,270) INTOD2,INTWP2,INTBK2,INTRQ2,INTFL2,FILP2
           WRITE(6,330)ITORD(5),ITWFP(5),ITBAKD(5),ITREQ(5),ITFILL(5),FIL(5)
          330 FORMAT(///,"OVRHL LRU",5I10,F10.3)
           WRITE(6,340) ITORD(6),ITWFP(6),ITBAKD(6),ITREQ(6),ITFILL(6),FIL(6)
          340 FORMAT(//,"OVRHL SRU",5I10,F10.3)
           WRITE(6,270)INTOD3,INTWP3,INTBK3,INTRQ3,INTFL3,FILP3
            WRITE(6,320)IGDORD,IGDWFP,IGDBAK,IGDREQ,IGDFIL,FILGD
          320 FORMAT(///,"GRAND TOT", 5I10,F10.3/
     &             " ",13X,6("=========",2X)/
```

```
OR 01  10-30-79   10.275   .S  RIMR 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      &         /T16,3("IBODAT",18X),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(I5,12I10)
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
```

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

```
*#RUN=IRIME/OBS/OUT2.8(BCD,NOGO)
*OUT2.S
          SUBROUTINE OUT2(KF,KL)
C                   THIS ROUTINE  OUTPUTS SUMMARY STATISTICS FOR STATISTICS
C                   CATEGORIES KF THRU KL, WHERE
C                           1=LRU'S AT BKSB
C                           2=SRU'S AT BKSB
C                           ETC.
           DIMENSION ITOTL(29 ),AVEYR(29 )
          DIMENSION TEXT(5,6)
C
          COMMON/NLEM/NLAM,MLAM
          COMMON/NREPL/NREPL,MREPL
          COMMON/NGROUP/NGROUP
C
            COMMON/INOTR/INOTR
           COMMON/ITIME/ITIME
           COMMON/ITYEAR/ITYEAR
           COMMON/ITINV/ITINV
           COMMON/INVACT/INVACT(1)
           COMMON/NDENT/NDENT(1)
C
           COMMON/NITEM/NITEM
           COMMON/IBOP/IBOPON(3),IBOPOM(8)
           COMMON/IBACKI/IBACKI(16,3,6)
           COMMON/IBACKT/IBACKT(16,3,6)
           COMMON/IBAKDI/IBAKDI(16,3,6)
           COMMON/IBAKDT/IBAKDT(16,3,6)
           COMMON/ICANCL/ICANCL(16,3,6)
           COMMON/IDISPS/IDISPS(16,3,6)
           COMMON/IEXPED/IEXPED(16,3,6)
           COMMON/IFILLI/IFILLI(16,3,6)
           COMMON/IFILLT/IFILLT(16,3,6)
           COMMON/IORDER/IORDER(16,3,6)
           COMMON/INVDAY/INVDAY(16,3,6)
           COMMON/INVOH/INVOH(16,3,6)
           COMMON/INVOR/INVOR(16,3,6)
           COMMON/IRATON/IRATON(16,3,6)
           COMMON/IRECET/IRECET(16,3,6)
           COMMON/IREOC /IREOC(16,3,6)
           COMMON/IREOI /IREOI(16,3,6)
           COMMON/IREOT/IREOT(16,3,6)
           COMMON/IRETRN/IRETRN(16,3,6)
           COMMON/ISHIPI/ISHIPI(16,3,6)
           COMMON/ISHIPT/ISHIPT(16,3,6)
           COMMON/ITERM /ITERM(16,3,6)
C------------------------------REPARABLE COMMONS
          COMMON/IREPGN/IREPGN(16,3,6)
           COMMON/ICNDEM/ICNDEM(16,3,6)
           COMMON/INRTS/INRTS(16,3,6)
           COMMON/IRECPL/IRECPL(16,3,6)
           COMMON/IWIP/IWIP(16,3,6)
```

```
                COMMON/IWFP/IWFP(16,3,8)
      C-------------------------
      C
      C
                 DATA TEXT/"LRU ","AT B","ASE ","FACI","LITY",
      &                    "SRU ","AT B","ASE ","FACI","LITY",
      &                    "    ","LRU ","AT D","EPOT","    ",
      &                    "    ","SRU ","AT D","EPOT","    ",
      &                    "  LR","U IN"," OVE","RHAU","L   ",
      &                    "  SR","U IN"," OVE","RHAU","L   "/
      C
      C           LOOP THROUGH LOCATION CATEGORIES K
      C
              R=FLOAT(ITIME)/FLOAT(IYEAR)
      C
      C       PRINT REPORT HEADER
      C
              WRITE(6,100)
              WRITE(6,102)
              WRITE(6,103)
              WRITE(6,102)
            1 CONTINUE
          DO 560 K=KF,KE
      C
      C
         60 WRITE(6,122)(TEXT(KK,K),KK=1,5)
              WRITE(6,123)NGROUP,NREPL,NLAM
          123 FORMAT(T8C,"NGROUP =",I3,"    NREPL =",I3,
      &                  "    NLAM =",I3)
      C
      C
      C           WRITE OUT COLUMN HEADINGS
      C
              WRITE(6,108)
          DO 260 J=1,3
              JJ=J
              CALL STATRD(JJ,ITOTL)
               DO 50 I=1,INOTB
               ITOTL(1)=ITOTL(1)+INVOH(I,J,K)
               ITOTL(2)=ITOTL(2)+INVOR(I,J,K)
               ITOTL(3)=ITOTL(3)+IRECET(I,J,K)
               ITOTL(4)=ITOTL(4)+IRETRN(I,J,K)
               ITOTL(5)=ITOTL(5)+ISHIPT(I,J,K)
               ITOTL(6)=ITOTL(6)+ISHIPI(I,J,K)
               ITOTL(7)=ITOTL(7)+IORDER(I,J,K)
               ITOTL(8)=ITOTL(8)+IREQT(I,J,K)
               ITOTL(9)=ITOTL(9)+IREQC(I,J,K)
               ITOTL(10)=ITOTL(10)+IREQI(I,J,K)
          50 WRITE(6,120)I,INVOH(I,J,K),INVOR(I,J,K),IRECET(I,J,K),
      &        IRETRN(I,J,K),
      &        ISHIPT(I,J,K),ISHIPI(I,J,K),IORDER(I,J,K),IREQT(I,J,K),
      &        IREQC(I,J,K),IREQI(I,J,K)
```

```
          WRITE(6,110)
          WRITE(6,202) (ITOTL(I),I=1,10)
    202 FORMAT(1H ,8H*TOTALS*,T20,10I10)
C
C                  COMPUTE AVERAGES
C
          DO 203 I=1,10
    203 AVEYR(I)=FLOAT(ITOTL(I))/R
          WRITE(6,204) (AVEYR(I),I=1,10)
    204 FORMAT(1H ,6HAVE/YR,T21,10F10.0)
    260 CONTINUE
          WRITE(6,122)(TEXT(KK,K),KK=1,5)
          WRITE(6,123)NGROUP,NREPL,NLKM
          WRITE(6,111)
   DO 360 J=1,3
            JJ=J
          CALL STATMD(JJ,ITOTL)
C
C                  ACCUMULATE OVER ALL QUARTERS
C
            DO 57 I=1,IMQTR
          ITOTL(11)=ITOTL(11)+IEXPED(I,J,K)
          ITOTL(12)=ITOTL(12)+IRATON(I,J,K)
          ITOTL(13)=ITOTL(13)+IDISPS(I,J,K)
          ITOTL(14)=ITOTL(14)+ITERM(I,J,K)
         ITOTL(24)=ITOTL(24)+IREPGN(I,J,K)
         ITOTL(25)=ITOTL(25)+ICNDEM(I,J,K)
         ITOTL(26)=ITOTL(26)+INRTS(I,J,K)
         ITOTL(27)=ITOTL(27)+IRECPL(I,J,K)
         ITOTL(28)=ITOTL(28)+IWIP(I,J,K)
         ITOTL(29 )=ITOTL(29)+IWFP(I,J,K)
     57 WRITE(6,120) I,IEXPED(I,J,K),IRATON(I,J,K),IDISPS(I,J,K),
   & ITERM(I,J,K),IREPGN(I,J,K),ICNDEM(I,J,K),INRTS(I,J,K),IRECPL(I,J,K),
   &             IWIP(I,J,K),IWFP(I,J,K)
          WRITE(6,67)
         WRITE(6,205)(ITOTL(I),I=11,14),(ITOTL(I),I=24,29)
    205 FORMAT(T12,'*TOTAL*',T20,10I10)
C
C                  COMPUTE AVERAGES
C
          DO 206 I=11,14
    206 AVEYR(I)=FLOAT(ITOTL(I))/R
          DO 220 I=24,29
   220 AVEYR(I)=FLOAT(ITOTL(I))/R
          WRITE(6,207) (AVEYR(I),I=11,14),(AVEYR(I),I=24,29)
    207 FORMAT(T13,'AVE/YR',T21,10F10.0)
   360 CONTINUE
C
C                  WRITE OUT COLUMN HEADINGS
C
         WRITE(6,122)(TEXT(KK,K),KK=1,5)
          WRITE(6,123)NGROUP,NREPL,NLKM
```

```
          WRITE(6,115)
          WRITE(6,68)
      DO 460 J=1,3
          JJ=J
        CALL STATED(JJ,ITOTL)
C                 DETERMINE NUMBER OF REQUISTIONS FROM CUSTOMERS
C                       EACH QUARTER
           DO 58 I=1,INQTR
          IF(IREQT(I,J,K).LE.0) GO TO 62
          T1=FLOAT(IFILLT(I,J,K))/FLOAT(IREQT(I,J,K))
          GO TO 63
       62 T1=0.
       63 CONTINUE
          IF(IREQI(I,J,K).LE.0) GO TO 64
          T2=FLOAT(IFILLI(I,J,K))/FLOAT(IREQI(I,J,K))
          GO TO 65
       64 CONTINUE
          T2=0.
       65 CONTINUE
          ITOTL( 8)=ITOTL(8)+IREQT(I,J,K)
          ITOTL(10)=ITOTL(10)+IREQI(I,J,K)
          ITOTL(15)=ITOTL(15)+IBACKT(I,J,K)
          ITOTL(16)=ITOTL(16)+IBACKI(I,J,K)
          ITOTL(17)=ITOTL(17)+IBAKDT(I,J,K)
          ITOTL(18)=ITOTL(18)+IBAKDI(I,J,K)
          ITOTL(19)=ITOTL(19)+INVDAY(I,J,K)
          ITOTL(20)=ITOTL(20)+IFILLT(I,J,K)
          ITOTL(21)=ITOTL(21)+IFILLI(I,J,K)
       58 WRITE(6,117) I,IBACKT(I,J,K),IBACKI(I,J,K),IBAKDT(I,J,K),
     &     IBAKDI(I,J,K),INVDAY(I,J,K),IFILLT(I,J,K),IFILLI(I,J,K),T1,T2
          WRITE(6,68)
          WRITE(6,208) (ITOTL(I),I=15,21)
      208 FORMAT(1H0,8X,8H"TOTALS",1X,7I10,2X)
          DO 209 I=15,21
      209 AVEYR(I)=FLOAT(ITOTL(I))/N
         AVEYR(22)=0.
         AVEYR(23)=0.
          IF(ITOTL(8).GT.0)AVEYR(22 )=FLOAT(ITOTL(20))/FLOAT(ITOTL(8))
        IF(ITOTL(10).GT.0)AVEYR(23 )=FLOAT(ITOTL(21))/FLOAT(ITOTL(10))
      210 FORMAT( 9X,7HAVE/YR  ,7F12.0,2F14.3)
          WRITE(6,210) (AVEYR(I),I=15,23)
      460 CONTINUE
C
C                 INCREMENT AGGREGATE CATEGORY
C
      560 CONTINUE
      900 RETURN
                67 FORMAT( T12,11(8(1H-),2X))
       68 FORMAT(1H ,11X,2H--,2X,7(12H   ---------),2(14H     ----------))
      100 FORMAT(1H1,/,/,/,/)
      101 FORMAT(1H ,/,/)
      102 FORMAT(1H ,48X,44(1H=))
```

```
31T 01  10-20-79    1..551

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( T22,10(8(1H-),2X))
211          *
212          108 FORMAT(//,
213         & T22,'INVENTORY INVENTORY',T64,'TOTAL     PRI I',
214         &      T83,'ORDERS     TOTAL     REQS   PRIORITY I'/
215         &      T13,'PERIOD    ON-HAND   ONORDER RECEIPTS',
216         &      T53,'RETURNS  SHIPMENTS SHIPMENTS  PLACED',
217         &      T95,'REQS   CANCELED   REQS'/
218         &     T12,11(8(1H-),2X))
219          111 FORMAT(//,
220         &     T32,'RATIONING',
221         &     T92,'COMPLETED EOP WORK WAIT FOR'/
222         &     T13,'PERIOD   EXPEDITES  ACTIONS',
223         &     T42,'DISPOSALS  TERMIN   REP GENS',
224         &      T72,'CONDEMNS   NRTS      REPAIRS IN',
225         &     T104,'PROCESS   PARTS'/
226         &     T12,11(8(1H-)2X))
227          115 FORMAT(//,
228         &     T21,'TOTAL       PRIORITY I      TOTAL ',
229         &     T55,'     PRI I     INVENTORY      TOTAL',
230         &     T93,'PRIORITY I  TOT FILLS   PRI I FILLS'/
231         &     T13,'PERIOD   BACKORDERS  BACKORDERS',
232         &     T45,' BO-DAYS     BO-DAYS     WEEKS',
233         &     T84,'FILLS        FILLS    /TOT REQ    /PRI I REQS')
234          117 FORMAT(1H ,T13,I2,T17,7(2X,I8),3X,2(1X,F10.2))
235          120 FORMAT(1H ,T13,I2,T20,10(2X,I8))
236          121 FORMAT(1H ,1X,3HBOP,2X,2(8X,I8))
237          122 FORMAT(1H1,59X,5A4/1H ,56X,20A1H*))
238               END
***W     7 MEMORY EXPANDED. USE $LIMITS OR CORE= OPTION FOR NEXT RUN
```

Subroutine: RCVPRT

Function:

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.

```
      SUBROUTINE RCVPRT(N,IQTY,NJOB)
C     EVENT CODE 17. RECORD RECEIPT OF IQTY UNITS OF ITEM N
C     TO SUPPORT REPAIR FOR NJOB. IF ALL PARTS NEEDS ARE
C     NOW SATISFIED, SCHEDULE BEGINNING OF REPAIR (EVENT
C     CODE 18) FOR THIS ITEM.
C
      COMMON /GCOM1/ ATRIB(30),JEVNT,MFA,MFE(100),MLE(100),MSTOP,NCRDR,
     & NNAPC,NNAPT,NNATR,NNFIL,NNQ(100),NNTRY,NPRNT,PPARM(50,4),TNOW,TTBEG,
     &  TTCLR,TTFIN,TTRIB(30),TTSET
      COMMON /GCOM6/ EENQ(100),IINN(100),KKRNK(100),MMAXQ(100),
     &  QQTIM(100),SSOBV(25,5),SSTPV(25,6),VVNC(100)
      DIMENSION NSET(1)
      COMMON QSET(1)
      EQUIVALENCE (NSET(1),QSET(1))
C
      COMMON/IDBUG/IDBUG
      COMMON/ITDAY/ITDAY
      COMMON/ITIME/ITIME
      COMMON/ITWEEK/ITWEEK
      COMMON/IWFP/IWFP(16,3,6)
      COMMON/IDORT/IDORT(1)
      COMMON/IDRT/IDRT(1)
      COMMON/IBRT/IBRT(1)
C
      IF(IDBUG.NE.1) GO TO 20
      WRITE(6,33)ITIME,N,IQTY,NJOB
   20 CONTINUE
C
C     LOCATE ENTRY NUMBER NTRY OF JOB NJOB IN THE BACKORDER FI
C     (FILE 2). JOB NUMBERS ARE RECORDED IN ATTRIBUTE 3
C
      XVAL=NJOB
      NTRY= NFIND(XVAL,5,2,3,.2)
      IF(NTRY.GT.0)GO TO 100
C
C     IF NTRY IS ZERO, NO ENTRY WAS FOUND.
C     PRINT AN ERROR MESSAGE, AT RETURN.
C
      WRITE(6,33) ITIME,N,IQTY,NJOB
   33 FORMAT("*****RCVPRT    ITIME=",I8,"  N=",I8,"  IQTY=",I8,
     &  "  NJOB=",I8)
      WRITE(6,13)"----------NO PARTS RECORD FOR NJOB WAS FOUND"
   13 FORMAT(V)
      RETURN
C
C     DECREASE PARTS NEEDS TO REFLECT THIS RECEIPT.
C     NEEDS ARE STORED IN ATTRIBUTE 2.
C
  100 CONTINUE
      NAT2=NTRY+2

      QSET(NAT2)=QSET(NAT2)-IQTY
```

```
      IF(IDBUG.EQ.1)WRITE(6,13)"----PARTS STILL NEEDED=",QSET(NAT2)
C
C                  IF PARTS ARE STILL NEEDED, RETURN
C
      IF(QSET(NAT2).GT.0.)RETURN
C
C                  ALL PARTS NEEDS HAVE BEEN SATISFIELD.
C                  REMOVE THIS JOB FROM WAIT-FOR-PARTS FILE.
C
      CALL RMOVE(NTRY,2)
      N=ATRIB(1)
      NJOB=ATRIB(3)
      IWAIT=(ITIME-ATRIB(4))
C
C                  RECORD TIME WAITING FOR PARTS
C
      CALL CUM(IWFP,IWAIT,N)
C
C                  DETERMINE IF THIS REPAIR SITE IS A DEPOT(1 OR 2),A
C                  BASE (3 OR 4), OR OVERHAUL LOCATION(5 OR 6)
C
      JTIME=0
      NCODE=KNSKU(N)
C
C                 IF DEPOT
C
      IF ((NCODE.EQ.5).OR.(NCODE.EQ.6))JTIME=IDORT(N)*ITDAY
C
C                  IF A BASE LOCATION
C
            IF((NCODE.EQ.3).OR.(NCODE.EQ.4))JTIME=IBRT(N)*ITDAY
C
C                  IF AN OVERHAUL FACILITY
C
      IF((NCODE.EQ.1).OR.(NCODE.EQ.2))JTIME=IDRT(N)*ITDAY
C
C                 SCHEDULE REPAIR COMPLETION
C
      JTIME=JTIME + ITIME
      CALL ENTER(JTIME,18,N,IOTY,NJOB)
      RETURN
      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=;RIME/OBJ/RECEIV.O(BCD,NOGO)
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/ISHIPI/ISHIPI(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(1)
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 FILIBO(N)
31            RETURN
32            END
```

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 bookeeping process, see Volume I, Section II.

```
      SUBROUTINE REMOVE(KTIME,KTYPE,KFSN,KQTY,KPRIOR)
C       THIS ROUTINE REMOVES THE EARLIEST TRANSACTION FROM THE
C       CURRENT EVENTS CHAIN,AND UPDATES THE CHAIN STRUCTURE
      COMMON/IDBUG/IDBUG
      COMMON/NENTRY/NENTRY
      COMMON/NFIRST/NFIRST
      COMMON/NTIME/NTIME
      COMMON/ILOCFE/ILOCFE(1)
      COMMON/JFSN/JFSN(1)
      COMMON/JPOINT/JPOINT(1)
      COMMON/JPRIOR/JPRIOR(1)
      COMMON/JQTY/JQTY(1)
      COMMON/JTIME/JTIME(1)
      COMMON/JTYPE/JTYPE(1)
      K=NFIRST
      KTIME=JTIME(K)
      KTYPE=JTYPE(K)
      KFSN=JFSN(K)
      KQTY=JQTY(K)
      KPRIOR=JPRIOR(K)
      NFIRST=JPOINT(K)
      NTIME=JTIME(NFIRST)
      IF(IDBUG.NE.1) GO TO 25
      WRITE(6,8000)KTIME,KTYPE,KFSN,KQTY,KPRIOR,K
 8000 FORMAT(7H REMOVE,I10,5I8)
   25 CONTINUE
C       IS THIS THE LAST TRANSACTION ON THE LIST
      IF(NENTRY.GT.1) GO TO 19
C       SET LIST CLOCK UP A LARGE INCREMENT
      NTIME=9999999
C       RECORD LOCATION K AS AVAILABLE
   19 ILOCFE(NENTRY)=K
C       DECREMENT COUNT OF LIST ENTRIES
      NENTRY=NENTRY-1
      RETURN
      END
```

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.

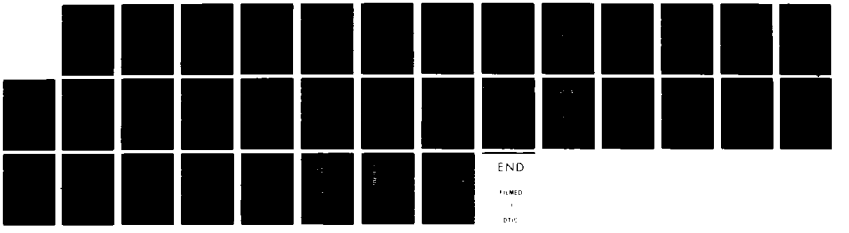
Description:

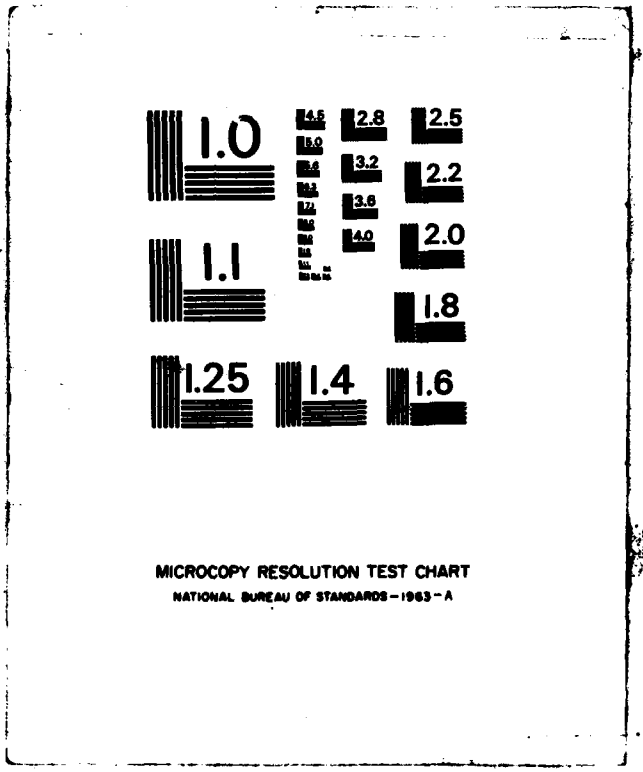
The routine first calls subroutine CUM to update the reparable generation statistics array IREPGN. Work-in-process for SKU N (INWIP(N)) is then increased by IQTY units, and logic returns to the calling program.

```
01  10-20-79    10,300    ,S
      **RUN=;RIME/OBJ/REPAIR,O(BCD,NOGO)
      *REPAIR,S
                      SUBROUTINE REPGEN(N,IQTY,NJOB)
      C                EVENT CODE 14. THIS ROUTINE REGARDS THE GENATION OF
      C                IQTY UNITS OF SKU N. NJOB IS THE NUMBER ASSIGNED TO
      C               THIS REP GEN.
      C
            COMMON/IREPGN/IREPGN(16,3,6)
             COMMON/INWIP/INWIP(1)
      C
      C                  UPDATE REP GENS STATISTICS
      C
            CALL CUM(IREPGN,IQTY,N)
      C
      C               UPDATE WORK-IN-PROCESS
      C
            INWIP(N)=INWIP(N)+   IQTY
            RETURN
            END
```

AD-A121 987 RIME: THE RECOVERABLE ITEM MANAGEMENT EVALUATOR VOLUME II SECTION III RIM.. (U) DECISION SYSTEMS DAYTON OH W S DEMMY MAY 80 TR-80-02-C F33600-78-C-0524 2/2

UNCLASSIFIED F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
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 Parameterrs:

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 = 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.

```
*@RUN=,RIMR/OBJ/REQ.O(BCD,NOGO)
      SUBROUTINE REQ(N,IQTY,LPRI,LTIME)
C           THIS ROUTINE REFLECTS PROCESSING OF A REQUISITION FOR IQTY
C           UNITS OF ITEM N DURING        PERIOD I,WHERE IPRI=1 DENOTES A H
C           PRIORITY REQUISITION, IPRI=2 DENOTES A LOW PRIORITY REQUISITIO
C           JTIME DENOTES THE CLOCK TIME THE REQ WAS RECEIVED
C            RECORD REQUISITION DATA.
C               IF THIS REQUISITION IS TO PROVIDE PARTS FOR LRU
C                 REPAIR, LPRI EQUALS (100*NJOB + PRIORITY CODE).
C
C           RECORD        REQUISITION DATA
       COMMON/ITIME/ITIME
       COMMON/ITDAY/ITDAY
       COMMON/LTPROD/LTPROD(1)
      COMMON/REQSIZ/REQSIZ(1)
      COMMON/REQMAD/REQMAD(1)
      COMMON/NDEMAC/NDEMAC(1)
       COMMON/NREQAC/NREQAC(1)
      COMMON/IREQI /IREQI(1)
      COMMON/IREQT/IREQT(1)
      COMMON/INVACT/INVACT(1)
      COMMON/ISUL/ISUL(1)
      COMMON/ISHIPI/ISHIPI(1)
      COMMON/ISHIPT/ISHIPT(1)
      COMMON/IFILLT/IFILLT(1)
      COMMON/IFILLI/IFILLI(1)
C
C            COMPUTE THE PRIORITY CODE AND STOCK KEEPING NUMBER
C                FOR THIS REQUISITION.
C
     IPRI=MOD(LPRI,100)
     NSKU=(LPRI-IPRI)/100
    IF(NSKU.LE.0) NSKU=N
C
      CALL CUM(IREQT,IQTY,N)
C           IS THIS A PRIORITY 1 REQ
      IF(IPRI.NE.1) GO TO 20
      CALL CUM(IREQI,IQTY,N)
C           UPDATE        DEMAND ACCUMULATOR AND EXPONENTIALLY SMOOTHED
   20 NDEMAC(N)=NDEMAC(N)+IQTY
      NREQAC(N)=NREQAC(N)+1
C           IS THERE ANY STOCK ON HAND
      IF(INVACT(N).GT.0) GO TO       40
C           PUT THIS REQUISITION IN BACKORDER STATUS
      CALL ENTERB(N,IQTY,LPRI,LTIME)
      RETURN
C           SET ITEST=STOCK REMAINING IF THE ORDER WERE FILLED COMPLETELY
   40 ITEST=INVACT(N)-IQTY
C           IS ITEST ABOVE THE SUPPORT LEVEL
      IF(ITEST.GE.ISUL(N)) GO TO 80
C           IS THIS REQ PRIORITY 1
      IF(IPRI.NE.1) GO TO 60
```

```
      C          CAN THE REQUISITION BE       COMPLETELY SATISFIED FROM STOCK O
           IF(INVACT(N).GE.IQTY) GO       TO 80
      C          SHIP ALL REMAINING ON-HAND STOCK IN PARTIAL FULFILLMENT
      C          OF THIS REQUISITION
           IQTYS=INVACT(N)
      C          COMPUTE QUANTITY TO BE       BACKORDERED AND UPDATE PRI-1 STAT
           IBK=IQTY-IQTYS
           CALL CUM(ISHIPI,IQTYS,N)
           CALL CUMB(IFILLI,IQTYS,N)
           GO TO 65
      C          ARE ON-HAND ASSETS ABOVE THE SUPPORT LEVEL
        60 IF(INVACT(N).LE.ISUL(N))       GO TO 70
      C          SHIP DOWN TO THE SUPPORT LEVEL
           IQTYS=INVACT(N)-ISUL(N)
      C          BACKORDER THE REMAINING QUANTITY
           IBK=IQTY-IQTYS
        65   IF(IBK.GT.0)CALL ENTERB(N,IBK,LPRI,LTIME)
      C          UPDATE       STOCK STATUS RECORDS
           INVACT(N)=INVACT(N)-IQTYS
      C
      C            IF NSKU> 1000, SCHEDULE A PARTS RECEIPT EVENT,
      C               (EVENT CODE 17)
      C
           JTIME=ITIME + 10
          IF(NSKU.GT.1000)CALL ENTER(JTIME,17,N,IQTYS,NSKU)
      C
      C            IF REQUISITION IS TO REPLENISH A LOWER LEVEL,
      C              SCHEDULE A RECEIVE EVENT(EVENT CODE 2).
      C
          JTIME=ITIME + LTPROD(NSKU)*ITDAY
          IF((NSKU.NE.N).AND.(NSKU.LT.1000))
     &          CALL ENTER(JTIME,2,NSKU,IQTYS,N)
      C
      C          UPDATE       GROSS PERFORMANCE STATISTICS
           JQTYS=-IQTYS
           CALL CUM(ISHIPT,IQTYS,N)
           CALL CUMB(IFILLT,IQTYS,N)
           RETURN
        70 CONTINUE
      C          BACKORDER ENTIRE REQUISITION
           CALL ENTERB(N,IQTY,LPRI,LTIME)
           RETURN
      C          SHIP TO FILL THE ENTIRE REQUISITION
        80 IQTYS=IQTY
      C          UPDATE       THE STOCK STATUS RECORDS
           INVACT(N)=INVACT(N)-IQTYS
      C
      C            IF NSKU> 1000, SCHEDULE A PARTS RECEIPT EVENT,
      C               (EVENT CODE 17)
      C
           JTIME=ITIME + 10
          IF(NSKU.GT.1000)CALL ENTER(JTIME,17,N,IQTYS,NSKU)
```

```
      C
      C          IF REQUISITION IS TO REPLENISH A LOWER LEVEL,
      C              SCHEDULE A RECEIVE EVENT(EVENT CODE 2).
      C
            JTIME=ITIME + LTPROD(NSKU)*YTDAY
            IF((NSKU.NE.N).AND.(NSKU.LT.1000))
     &           CALL ENTER(JTIME,2,NSKU,IQTYS,N)
      C          UPDATE        PERFORMANCE
            JQTYS=IQTYS
            CALL CUM(ISHIPT,IQTYS,N)
            CALL CUM(IFILLT,IQTYS,N)
      C          IS THIS A PRIORITY 1 REQUISITION
            IF(IPRI.NE.1) GO TO 85
      C          UPDATE        PRIORITY 1 STATISTICS
            CALL CUM(ISHIPI,IQTYS,N)
            CALL CUM(IFILLI,IQTYS,N)
         85 CONTINUE
            RETURN
            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.

```
*#RUN=;RIME/OBJ/RIME.Q(BCD,NOGO)
*RIME.S
      CHARACTER TEXT*40
C
C           SET DIMENSIONS FOR STATISTICS ARRAYS
C
C
C--------------------------------------DEFINE GASP COMMONS-----
C
      DIMENSION NSET(1)
     COMMON QSET(600)
      EQUIVALENCE(NSET(1),QSET(1))
     COMMON /GCOM1/ ATRIB(30),JEVNT,MFA,MFE(100),MLE(100),MSTOP,NCRDR,
  &  NNAPO,NNAPT,NNATR,NNFIL,NNQ(100),NNTRY,NPRNT,PPARM(50,4),TNOW,TTBEG,
  &   TTCLR,TTFIN,TTRIB(30),TTSET
     COMMON /GCOM6/ EENQ(100),IINN(100),KKRNK(100),MMAXQ(100),
  &   QQTIM(100),SSOBV(25,5),SSTPV(25,6),VVNQ(100)
C
*-----------------------------------------------------------
     COMMON/G2ENT/ITWRT,IOUT,IGRAPH,ITSUMRY
     COMMON/IDBUG/IDBUG
      COMMON/IEBUG/IEBUG
     COMMON/IGBUG/IGBUG
     COMMON/IPBUG/IPBUG
      COMMON/INBUG/INBUG
     COMMON/IDDIV/IDDIV
     COMMON/IDLEVL/IDLEVL
     COMMON/IDSTAT/IDSTAT
     COMMON/IDTNQ/IDTNQ
     COMMON/INDEM/INDEM
     COMMON/INOTR/INOTR
     COMMON/IPFOR/IPFOR
     COMMON/IQTRND/IQTRND
     COMMON/ISTAT/ISTAT
     COMMON/ISTOCK/ISTOCK,IDSTOC
     COMMON/ISTOP/ISTOP
     COMMON/ITCAWB/ITCAWB,IDCAWB
     COMMON/ITDAY/ITDAY
     COMMON/ITDIV/ITDIV
     COMMON/ITFOR/ITFOR,IDFOR
     COMMON/ITNQ/ITNQ
     COMMON/ITINV/ITINV
     COMMON/ITIME/ITIME
     COMMON/ITLEVL/ITLEVL
     COMMON/ITMNTH/ITMNTH
     COMMON/ITQTR/ITQTR
     COMMON/ITWEEK/ITWEEK
     COMMON/ITYEAR/ITYEAR
     COMMON/NDMAX/NDMAX
      COMMON/NDEM/NDEM
      COMMON/NDNIS/NDNIS
     COMMON/NENTRY/NENTRY
```

```
              COMMON/NFEMAX/NFEMAX
              COMMON/NFIRST/NFIRST
              COMMON/NITEM/NITEM
              COMMON/NLOC/NLOC
              COMMON/NLOCBK/NLOCBK
              COMMON/NTIME/NTIME
      C----------------SIMULATION CONTROL PARAMETERS
             COMMON/IDENT/IDENT
             COMMON/NLAM/NLAM,MLAM
             COMMON/NREPL/NREPL,MREPL
             COMMON/NGROUP/NGROUP
             COMMON/NFGRP /NFGRP
             COMMON/NLGRP /NLGRP
      C------------------------REPAIR SYSTEM VARIABLES
              COMMON/NJOB/NJOB
              COMMON/NBASES/NBASES
              COMMON/NSRU/NSRU
              COMMON/NNQTR/NNQTR
      C------------------------------
              COMMON/INVACT/INVACT(40)
              COMMON/INVDUE/INVDUE(40)
             COMMON/INWIP/INWIP(40)
             COMMON/IBRT/IBRT(40)
             COMMON/IDRT/IDRT(40)
             COMMON/IDONT/IDONT(40)
              COMMON/IRL/IRL(40)
              COMMON/IROL/IROL(40)
              COMMON/IROTY/IROTY(40)
              COMMON/ISUL/ISUL(40)
              COMMON/ITL/ITL(40)
              COMMON/LTADM/LTADM(40)
              COMMON/LTPROD/LTPROD(40)
              COMMON/NBOPT/NBOPT(40)
               COMMON/NBOIU/NBOIU(40)
               COMMON/NBOIR/NBOIR(40)
               COMMON/NBOTR/NBOTR(40)
              COMMON/NBOTU/NBOTU(40)
              COMMON/NDEMAC/NDEMAC(40)
               COMMON/NRETAC/NRETAC(40)
               COMMON/NREQAC/NREQAC(40)
               COMMON/NDEMND/NDEMND(1,24)
               COMMON/NRETUR/NRETUR(1,24)
               COMMON/NREQ/NREQ(1,24)
              COMMON/NDENT/NDENT(40)
              COMMON/RNQMAD/RNQMAD(1)
              COMMON/RNQSIZ/RNQSIZ(1)
              COMMON/RNRSUM/RNRSUM(1)
              COMMON/RMAD  /RMAD(1)
              COMMON/RMEAN/RMEAN(1)
              COMMON/RNREOS/RNREOS(1)
              COMMON/RNTBR /RNTBR(1)
              COMMON/RSIGLT/RSIGLT(1)
```

```
      COMMON/ETREND/ETREND(1)
      COMMON/UCOST/UCOST(40)
      COMMON/ILOCPE/ILOCPE(500)
      COMMON/JPSN/JPSN(500)
      COMMON/JPOINT/JPOINT(500)
      COMMON/JPRIOR/JPRIOR(500)
      COMMON/JQTY/JQTY(500)
      COMMON/JTIME/JTIME(500)
      COMMON/JTYPE/JTYPE(500)
      COMMON/IBACPT/IBACPT(200)
      COMMON/IDPSNB/IDPSNB(200)
      COMMON/ILOCBK/ILOCBK(200)
      COMMON/IPRIOR/IPRIOR(200)
      COMMON/IQTYB/IQTYB(200)
      COMMON/ITMBAC/ITMBAC(200)
      COMMON/IBOP/IBOPOH(3),IBOPOO(3)
     COMMON/IBOPCT/IBOPSM(3),IBOPEG(3)
*********************************************
C             PERFORMANCE STATISTICS
C
C
C
      COMMON/IBACKI/IBACKI(16,3,6)
      COMMON/IBACKT/IBACKT(16,3,6)
      COMMON/IBAKDI/IBAKDI(16,3,6)
      COMMON/IBAKDT/IBAKDT(16,3,6)
     COMMON/IBODAT/IBODAT(16,3,6)
     COMMON/IBODAI/IBODAI(16,3,6)
      COMMON/ICANCL/ICANCL(16,3,6)
      COMMON/IDISPS/IDISPS(16,3,6)
      COMMON/IEXPED/IEXPED(16,3,6)
      COMMON/IFILLI/IFILLI(16,3,6)
      COMMON/IFILLT/IFILLT(16,3,6)
      COMMON/INVON/INVON(16,3,6)
      COMMON/INVDAY/INVDAY(16,3,6)
      COMMON/INVOR/INVOR(16,3,6)
      COMMON/IORDER/IORDER(16,3,6)
      COMMON/IRATON/IRATON(16,3,6)
      COMMON/IRECET/IRECET(16,3,6)
      COMMON/IRNOC /IRNOC(16,3,6)
      COMMON/IRNOI /IRNOI(16,3,6)
      COMMON/IRNOT/IRNOT(16,3,6)
      COMMON/IRETRN/IRETRN(16,3,6)
      COMMON/ISHIPI/ISHIPI(16,3,6)
      COMMON/ISHIPT/ISHIPT(16,3,6)
      COMMON/ITERM /ITERM(16,3,6)
     COMMON/ISHORD/ISHORD(16,3,6)
     COMMON/ILGORD/ILGORD(16,3,6)
C-----------------------------
     COMMON/IREPON/IREPON(16,3,6)
     COMMON/INRTS/INRTS(16,3,6)
     COMMON/ICUDEM/ICUDEM(16,3,6)
```

```
01  10-20-79   10.565

      COMMON/ERECPL/IRECPL(16,3,6)
      COMMON/IWIP/INIP(16,3,6)
      COMMON/IWIP/IWIP(16,3,6)
C-----*----*----*----*----*----*--
C
      COMMON/ITRACE/ITRACE,ISTRAC
C
C
C           REWIND EXOG. EVENT FILE
C
      REWIND 07
C           READ INPUT DATA
C
   10       CONTINUE
      WRITE(6,8000)
 8000      FORMAT(1H1, T30,'INVENTORY SYSTEM SIMULATOR'//T34,
     &     'RUN PARAMETERS'///)
C
C           READ EXOGENOUS EVENT FILE IDENTIFICATION RECORD
C
      READ(7)NFORP,NLORP,NBASES,NWOTR,NREPL
      WRITE(6,8001)NFORP,NLORP,NBASES,NWOTR,NREPL
 8001 FORMAT(///"      EXOGENOUS EVENT FILE CHARACTERISTICS"//
     &  " NFORP",T16,"FIRST LRU ORD =",T40,I5//
     &  " NLORP",T16,"LAST  LRU ORD =",T40,I5//
     &  " NBASES",T16,"NUMBER OF BASES =",T40,I5//
     &  " NWOTR",T16,"NUMBER OF QUARTERS =",T40,I5//
     &  " NREPL",T16,"NUMBER OF REPLICATIONS",T40,I5)
C
C           READ OUTPUT CONTROLS(CKRD CODE C2)
C
      READ(5,8003)ITWRT,IOUT,IGRAPH,ISUMRY
 8003      FORMAT(4)
      WRITE(6,8004)ITWRT,IOUT,IGRAPH,ISUMRY
 8004      FORMAT(///" (C2)  OUTPUT CONTROLS...(NOTE. 1=YES)"//
     &     ' ITWRT',T16,'IT.WRITE=',I3//
     &     ' IOUT',T16,'SUMMARY =',I3//
     &     ' IGRAPH',T16,'GRAPHS  =',I3//
     &     ' ISUMRY',T16,'SUMMARY =',I3)
C
C           READ DEBUG FLAGS
C
      READ(5,8005)IDBUG,IEBUG,IPBUG,IQBUG,INBUG,ITRACE,ISTRAC
      WRITE(6,8005)IDBUG,IEBUG,IPBUG,IQBUG,INBUG,ITRACE,ISTRAC
 8005      FORMAT(///" (C3)  DEBUG FLAGS"//
     &     T16,'IDBUG =',I3//
     &     T16,'IEBUG =',I3//
     &     T16,'IPBUG =',I3//
     &     T16,'IQBUG =',I3//
     &     T16,'INBUG =',I3///
     &     T16,'ITRACE=',I3//
     &     T16,'ISTRAC=',I3//)
```

```
 01   10-20-79    10.565

      C
      C            DEFINE ITEM INPUT FILES
      C
      C
      C            READ SIMULATION SIZE PARAMETERS(CARD TYPE C8)
      C
           READ(5,8003)NLAM,INQTR,NTOTL
           WRITE(6,8080)NLAM,INQTR,NTOTL
       8080         FORMAT(///' (C8)  SIMULATION SIZE'///
     &              ' NLAM',T16,'NUMBER OF LAMBDAS',T40,I5//
     &              ' INQTR',T16,'NUMBER OF QUARTERS',T40,I5//
     &              ' NTOTL',T16,'NO. OF LAU GROUPS',T40,I5//
     &                )
      C
      C            LIMIT INPUT PARAMETERS CONSISTENT WITH EXOG. FILE
      C
           IF(INQTR.GT.MNQTR)INQTR=MNQTR
           K=NLGRP*NFGRP+1
           IF(NTOTL.GT.K)NTOTL=K
      C
      C---INITIALIZE RANDOM NUMBER STREAM
      C
           RNLAST=RANDU(0,1)
      C
      C            REWIND THE LEVELS FILE (09)
      C
           REWIND 09
      C*****************************************************************
      C
      C---BEGIN LAGRANGIAN LOOP
      C
           DO 200  NLAM=1,NLAM
            IF(IDBUG.EQ.1)WRITE(6,8090)NLAM
      C
              809  FORMAT(//10('*****'),'BEGIN SIMULATION ',
           &       "FOR NLAM=",I4,10("*****")//)
      C
      C
      C              REWIND THE EVENTS FILE
      C            AND READ THE HEADER RECORD AGAIN
      C
           REWIND 07
           READ(7)NFGRP,NLGRP,NBASES,MNQTR,NREPL
      C
      C
      C            ZERO THE STATISTICS ARRAYS
      C
           CALL ZERO
      C
         100 CONTINUE
      C
```

```
 01  10-20-79   10.56$

      C            SIMULATE ALL NTOTL GROUPS, ONE GROUP AT A TIME
      C
               NITEM=1
            NGROUP=NPERP +1
              DO 400 N=1,NTOTL
                 NGROUP=NGROUP+1
                 WRITE(6,213)N
       213  FORMAT(1"0****BEGIN LRU GRP=",I5)
      C
      C---READ DATA FOR GROUP N
      C
              CALL INITN1
      C
      C---BEGIN REPLICATION LOOP
      C
           DO  390 NREPL=1,NREPL
      C
      C
      C
      C---SIMULATE THIS ITEM
      C
               CALL RVNTS(RNLAST)
      C
      C---RECORD RANDOM NUMBER SEED
      C
           RNLAST=RANDU(RNLAST)
            IF(ITWRT.NE.3)CALL ITRSLT
      C
      C            END OF REPLICATION LOOP(KREPL)
        390 CONTINUE
      C
      C            END OF GROUP LOOP
        400 CONTINUE
      C
      C
      C
      C----------OUTPUT PERFORMANCE STATISTICS
      C                  NOTE---OUTREP PARAMETER IS J (ACTIONS,UNITS,DOLLARS)
      C                       ---OUT2  PARAMETER IS KNSKU(BASE,DEPOT,OVERHAUL,
      C                            FOR SRU AND LRU.
      C
           IF(ISUMRY.EQ.1) CALL OUTREP(4,3)
          IF(IOUT.NE.0) CALL OUT2(1,5)
      C----------END OF LAGRANGIAN LOOP
       2000 CONTINUE
      C
      C---END OF JOB
       9999  CONTINUE
               STOP
               END
W     7 MEMORY EXPANDED. USE $LIMITS OR GO,M= OPTION FOR NEXT RUN
```

```
      *#RUN=;RIME/OBJ/RIMEB,O(BCD;NOGO)
      *RIMEB.S
            CHARACTER TEXT*40
      C
      C           SET DIMENSIONS FOR STATISTICS ARRAYS
      C
      C
      C-------------------------DEFINE GASP COMMONS-----
      C
            DIMENSION NSET(1)
           COMMON QSET(600)
            EQUIVALENCE(NSET(1),QSET(1))
           COMMON /GCOM1/ ATRIB(30),JEVNT,MFA,MFE(100),MLE(100),MSTOP,NCRDS,
     &   NNARO,NNAPT,NNATP,NNFIL,NNQ(100),NNTRY,NPRMT,PPARM(50,4),TNOW,ITBUG,
     &    TTCLR,TTFIN,TTRIB(80),TTSET
           COMMON /GCOM2/ EENQ(100),IINN(100),KKRNK(400),MMAXQ(100),
     &    QQTIM(100),SSOBV(25,5),SSTPV(25,6),VVNC(100)
      C
      *-------------------------------------------------
           COMMON/C2CNT/ITWRT,IOUT,IGRAPH,ISUMRY
            COMMON/IDBUG/IDBUG
            COMMON/IEBUG/IEBUG
            COMMON/IGBUG/IGBUG
            COMMON/IFBUG/IFBUG
            COMMON/IHBUG/IHBUG
            COMMON/IDDIV/IDDIV
            COMMON/IDLEVL/IDLEVL
            COMMON/IDSTAT/IDSTAT
            COMMON/IDTHQ/IDTHQ
            COMMON/IKDEM/IKDEM
            COMMON/INQTR/INQTR
            COMMON/IPFOR/IPFOR
            COMMON/IQTRND/IQTRND
            COMMON/ISTAT/ISTAT
            COMMON/ISTOCK/ISTOCK,IDSTOC
            COMMON/ISTOP/ISTOP
            COMMON/ITCANB/ITCANB,IDCANB
            COMMON/ITDAY/ITDAY
            COMMON/ITDIV/ITDIV
            COMMON/ITFOR/ITFOR,IDFOP
            COMMON/ITHQ/ITHQ
            COMMON/ITINV/ITINV
            COMMON/ITIME/ITIME
            COMMON/ITLEVL/ITLEVL
            COMMON/ITMNTH/ITMNTH
            COMMON/ITQTR/ITQTR
            COMMON/ITWEEK/ITWEEK
            COMMON/ITYEAR/ITYEAR
            COMMON/NBMAX/NBMAX
            COMMON/NDEM/NDEM
            COMMON/NDHIS/NDHIS
            COMMON/NENTRY/NENTRY
```

```
      COMMON/NFEMAX/NFEMAX
      COMMON/NFIRST/NFIRST
      COMMON/NITEM/NITEM
      COMMON/NLOC/NLOC
      COMMON/NLOCBK/NLOCBK
      COMMON/NTIME/NTIME
C-----------SIMULATION CONTROL PARAMETERS
      COMMON/IDENT/IDENT
      COMMON/NLAM/NLAM,MLAM
      COMMON/NREPL/NREPL,MREPL
      COMMON/NGROUP/NGROUP
      COMMON/NFGRP /NFGRP
      COMMON/NLGRP /NLGRP
C-----------REPAIR SYSTEM VARIABLES
      COMMON/NJOB/NJOB
      COMMON/NBASES/NBASES
      COMMON/NSRU/NSRU
      COMMON/NNQTR/NNQTR
C-----------
      COMMON/INVACT/INVACT(340)
      COMMON/INVDUE/INVDUE(340)
      COMMON/INNIP/INNIP(340)
      COMMON/IBRT/IBRT(340)
      COMMON/IDRT/IDRT(340)
      COMMON/IDORT/IDORT(340)
      COMMON/IRL/IRL(340)
      COMMON/IROL/IROL(340)
      COMMON/IROTX/IROTX(340)
      COMMON/ISUL/ISUL(340)
      COMMON/ITL/ITL(340)
      COMMON/LTADM/LTADM(340)
      COMMON/LTPROD/LTPROD(340)
      COMMON/NBOPT/NBOPT(340)
      COMMON/NBOIU/NBOIU(340)
      COMMON/NBOIR/NBOIR(340)
      COMMON/NBOTR/NBOTR(340)
      COMMON/NBOTU/NBOTU(340)
      COMMON/NDEMAC/NDEMAC(340)
      COMMON/NRETAC/NRETAC(340)
      COMMON/NREQAC/NREQAC(340)
      COMMON/NDEMBD/NDEMBD(1,24)
      COMMON/NRETUR/NRETUR(1,24)
      COMMON/NREQ/NREQ(1,24)
      COMMON/NDENT/NDENT(340)
      COMMON/REQMAD/REQMAD(1)
      COMMON/REQSIZ/REQSIZ(1)
      COMMON/RERSUM/RERSUM(1)
      COMMON/RMAD  /RMAD(1)
      COMMON/RMEAN/RMEAN(1)
      COMMON/RMREQS/RMREQS(1)
      COMMON/RMTBR /RMTBR(1)
      COMMON/RSIGLT/RSIGLT(1)
```

```
      COMMON/RTREND/PTREND(4)
      COMMON/UCOST/UCOST(800)
      COMMON/ILOCPE/ILOCPE(800)
      COMMON/JPSN/JPSN(500)
      COMMON/JPOINT/JPOINT(500)
      COMMON/JPRIOR/JPRIOR(500)
      COMMON/JOTY/JOTY(500)
      COMMON/JTIME/JTIME(500)
      COMMON/JTYPE/JTYPE(500)
      COMMON/IBACPT/IBACPT(800)
      COMMON/IDPSNB/IDPSNB(800)
      COMMON/ILOCBK/ILOCBK(800)
      COMMON/IPRIOR/IPRIOR(800)
      COMMON/IOTYB/IOTYB(800)
      COMMON/ITMBAC/ITMBAC(800)
      COMMON/IBOP/IBOPOH(3),IBOPOR(3)
     COMMON/IBOPCT/IBOPSM(3),IBOPLS(3)
*****************************************
C            PERFORMANCE STATISTICS
C
C
C
      COMMON/IBACKI/IBACKI(16,3,6)
      COMMON/IBACKT/IBACKT(16,3,6)
      COMMON/IBAKDI/IBAKDI(16,3,6)
      COMMON/IBAKDT/IBAKDT(16,3,6)
     COMMON/IBODAT/IBODAT(16,3,6)
     COMMON/IBODAI/IBODAI(16,3,6)
      COMMON/ICANCL/ICANCL(16,3,6)
      COMMON/IDISPS/IDISPS(16,3,6)
      COMMON/IEXPED/IEXPED(16,3,6)
      COMMON/IFILLI/IFILLI(16,3,6)
      COMMON/IFILLT/IFILLT(16,3,6)
      COMMON/INVOH/INVOH(16,3,6)
      COMMON/INVDAY/INVDAY(16,3,6)
      COMMON/INVOR/INVOR(16,3,6)
      COMMON/IORDER/IORDER(16,3,6)
      COMMON/IRATOR/IRATOR(16,3,6)
      COMMON/IRECET/IRECET(16,3,6)
      COMMON/IREQC /IREQC(16,3,6)
      COMMON/IREQI /IREQI(16,3,6)
      COMMON/IREQT/IREQT(16,3,6)
      COMMON/IRETRN/IRETRN(16,3,6)
      COMMON/ISHIPI/ISHIPI(16,3,6)
      COMMON/ISHIPT/ISHIPT(16,3,6)
      COMMON/ITERM /ITERM(16,3,6)
     COMMON/ISHORD/ISHORD(16,3,6)
     COMMON/ILGORD/ILGORD(16,3,6)
C-----------------------
      COMMON/IREPGN/IREPGN(16,3,6)
      COMMON/INRTS/INRTS(16,3,6)
      COMMON/ICNDEM/ICNDEM(16,3,6)
```

```
      COMMON/IRECPL/IPECPL(16,3,6)
      COMMON/INTP/IWIP(16,8,6)
      COMMON/IWPP/IWPP(16,8,6)
C-----*----*----*----*----*----*-
C
      COMMON/ITRACE/ITRAC,ISTRAC
C
C
C           REWIND EXOG. EVENT FILE
C
      REWIND 07
C           READ INPUT DATA
C
   10    CONTINUE
         WRITE(6,8000)
 8000     FORMAT(1H1, T30,'INVENTORY SYSTEM SIMULATOR'//T34,
     &     'RUN PARAMETERS'///)
C
C           READ EXOGENOUS EVENT FILE IDENTIFICATION RECORD
C
      READ(7)NFGRP,NLGRP,NBASES,NNQTR,NREPL
      WRITE(6,8001)NFGRP,NLGRP,NBASES,NNQTR,NREPL
 8001 FORMAT(///'       EXOGENOUS EVENT FILE CHARACTERISTICS'//
     &  ' NFGRP',T16,'FIRST LRU GRP =',T40,I5//
     &  ' NLGRP',T16,'LAST  LRU GRP =',T40,I5//
     &  ' NBASES',T16,'NUMBER OF BASES =',T40,I5//
     &  ' NNQTR',T16,'NUMBER OF QUARTERS =',T40,I5//
     &  ' NREPL',T16,'NUMBER OF REPLICATIONS',T40,I5)
C
C           READ OUTPUT CONTROLS(CARD CODE C2)
C
      READ(5,8003)ITWRT,IOUT,IGRAPH,ISUMRY
 8003     FORMAT(V)
      WRITE(6,8004)ITWRT,IOUT,IGRAPH,ISUMRY
 8004     FORMAT(///' (C2)  OUTPUT CONTROLS...(NOTE, 1=YES)'//
     &     ' ITWRT',T16,'IT WRITE=',I3//
     &     ' IOUT',T16,'SUMMARY =',I3//
     &     ' IGRAPH',T16,'GRAPHS  =',I3//
     &     ' ISUMRY',T16,'SUMMARY =',I3)
C
C           READ DEBUG FLAGS
C
      READ(5,8003)IDBUG,IEBUG,IFBUG,IGBUG,IHBUG,ITRACE,ISTRAC
      WRITE(6,8005)IDBUG,IEBUG,IFBUG,IGBUG,IHBUG,ITRACE,ISTRAC
 8005     FORMAT(///' (C3)  DEBUG FLAGS'//
     &     T16,'IDBUG =',I3//
     &     T16,'IEBUG =',I3//
     &     T16,'IFBUG =',I3//
     &     T16,'IGBUG =',I3//
     &     T16,'IHBUG =',I3///
     &     T16,'ITRACE=',I8//
     &     T16,'ISTRAC=',I8//)
```

```
      C
      C          DEFINE ITEM INPUT FILES
      C
      C
      C          READ SIMULATION SIZE PARAMETERS(CARD TYPE C8)
      C
           READ(5,8003)NLAM,INQTR,NTOTL
           WRITE(6,8080)NLAM,INQTR,NTOTL
       8080       FORMAT(////' (C8)  SIMULATION SIZE'///
     &           ' NLAM',T16,'NUMBER OF LAMBDAS',T40,I5//
     &           ' INQTR',T16,'NUMBER OF QUARTERS',T40,I5//
     &           ' NTOTL',T16,'NO. OF LRU GROUPS',T40,I5//
     &            )
      C
      C          LIMIT INPUT PARAMETERS CONSISTENT WITH EXOG. FILE
      C
           IF(INQTR.GT.NNQTR)INQTR=NNQTR
           K=NLGRP+NFGRP+1
           IF(NTOTL.GT.K)NTOTL=K
      C
      C---INITIALIZE RANDOM NUMBER STREAM
      C
           NNLAST=RANDU(-,1)
      C
      C          REWIND THE LEVELS FILE (09)
      C
           REWIND 09
      C
      C*************************************************************
      C
      C---BEGIN LAGRANGIAN LOOP
      C
           DO 2000 NLAM=1,NLAM
            IF(IDBUG.EQ.1)WRITE(6,8090)NLAM
      C
          8090 FORMAT(//10('*****'),'BEGIN SIMULATION ',
           "FOR NLAM=",I4,10("*****")//)
      C
      C
      C             REWIND THE EVENTS FILE
      C          AND READ THE HEADER RECORD AGAIN
      C
           REWIND 07
           READ(7)NFGRP,NLGRP,NBASES,NNQTR,NREPL
      C
      C
      C          ZERO THE STATISTICS ARRAYS
      C
           CALL ZERO
      C
         100 CONTINUE
      C
```

```
      C           SIMULATE ALL NTOTL GROUPS, ONE GROUP AT A TIME
      C
            NITEM=1
          NGROUP=NPGRP -1
           DO 400 N=1,NTOTL
             NGROUP=NGROUP+1
             WRITE(6,213)N
  213  FORMAT("*****BEGIN LRU GRP=",I5)
      C
      C---READ DATA FOR GROUP N
      C
           CALL INITM1
      C
      C----BEGIN REPLICATION LOOP
      C
        DO  390 MREPL=1,NREPL
      C
      C
      C
      C---SIMULATE THIS ITEM
      C
           CALL EVNTS(RNLAST)
      C
      C---RECORD RANDOM NUMBER SEED
      C
        RNLAST=RANDU(RNLAST)
         IF(ITWRT.EQ.1)CALL ITRSLT
      C
      C          END OF REPLICATION LOOP(KREPL)
  390 CONTINUE
      C
      C          END OF GROUP LOOP
  400 CONTINUE
      C
      C
      C
      C----------OUTPUT PERFORMANCE STATISTICS
      C              NOTE---OUTREP PARAMETER IS J (ACTIONS,UNITS,DOLLARS)
      C                  ---OUT2  PARAMETER IS KNSKU(BASE,DEPOT,OVERHAUL,
      C                       FOR SRU AND LRU.
      C
        IF(ISUMRY.EQ.1) CALL OUTREP(1,3)
       IF(IOUT.NE.0) CALL OUT2(1,5)
      C-------------END OF LAGRANGIAN LOOP
  2000 CONTINUE
      C
      C---END OF JOB
  9999  CONTINUE
           STOP
           END
*W    7 MEMORY EXPANDED, USE SLIMITS OR CORE= OPTION FOR NEXT RUN
```

Subroutine: REVIEW

Function:

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.

```
*RUN=IRINE/OBS/REVIEW.O(BCD,NOWOY
*REVIEW.S
      SUBROUTINE REVIEW(NN,IFLAG)
C           THIS ROUTINE COMPARES THE INVENTORY POSITION-
C      (ON-HAND + ON-ORDER + WORK-INPROCESS - BACKORDERS)
C      TO THE DESIRED STOCK LEVEL (IROL(N)).
C      IF THE INVENTORY POSITION IS BELOW THIS LEVEL,
C       AN ORDER FOR THE DEFICIENCY IS PLACED.
C
C      IF NN=0, ALL ITEMS ARE REVIEWED.  OTHERWISE, ONLY
C       ITEM NN IS REVIEWED.
C
C
C                    IF IFLAG=1, SCHEDULE FOR IMMEDIATE RECEIPT TO
C                                SIMULATE INITIAL PROVISIONING STOCK POSITIONING.
C                  OTHERWISE, SCHEDULE ORDERS FOR RECEIPT AFTER THE
C                                STANDARD ADM AND PROD LEADTIMES.
C                FOR STANDARD DELIVERIES, DELIVERY TIMES ARE COMPUTED
C                  BY SUBROUTINE ORDER.
C
C
      COMMON/IDBUG/IDBUG
      COMMON/ICDFOR/ICDFOR
       COMMON/ITIME/ITIME
       COMMON/IMNTH/IMNTH
      COMMON/NDMAX/NDMAX
      COMMON/NITEM/NITEM
      COMMON/NLOCBK/NLOCBK
       COMMON/INVACT/INVACT(1)
       COMMON/INVDUE/INVDUE(1)
      COMMON/INWIP/INWIP(1)
       COMMON/IROL/IROL(1)
       COMMON/IROTY/IROTY(1)
       COMMON/LTADM/LTADM(1)
       COMMON/LTPROD/LTPROD(1)
       COMMON/NBOTU/NBOTU(1)
      COMMON/NBOTR/NBOTR(1)
C
C           IF NN=0, REVIEW ALL ITEMS; OTHERWISE ONLY REVIEW ITEM NN"
C
      NF=NN
      NL=NN
      IF(NN.EQ.0)NF=1
      IF(NN.EQ.0)NL=NITEM
C
      DO 200 NNN=NF,NL
     N=NNN
C
*
      NRS=NBOTR(N)
      INVPOS=INVACT(N)+INVDUE(N)+INWIP(N)-NBOTU(N)
      IF(IDBUG.EQ.1)WRITE (6,8013)N,INVACT(N),INVDUE(N),
```

03T 01 10-20-79 10.350

```
      &         INHIT(N),INOTU(N),INVPOS,NNN,IROL(N)
       8013 FORMAT(4X,"****REVIEW---N=",I3,"  QH=",I5,"  DUE-IN=",I5,
      &         "  WIN=",I5,"  BO=",I5,"  INVPOS=",I5,"  IRO=SHORT=",I5,
      &      "  IROL=",I5)
      C
      C              COMPARE STOCK POSITION TO REORDER POINT
      C
            IF(INVPOS.GE.IROL(N)) GO TO 2000
      *
      *          COMPUTE STOCK LEVEL DEFICIENCY
      *
         97 CONTINUE
            IRO=IROL(N)-INVPOS
      *
      *      COMPUTE DELIVERY DATES     FOR ORDER
      *
        95  IF(IFLAG.EQ.1) GO TO 120
      *
      *      PLACE ORDER FOR IRO UNITS
      C              DELIVERY TIME IS COMPUTED BY SUBROUTINE ORDER.
      *
            CALL ORDER(N,IRO,0)
            GO TO 2000
      C
      C          SCHEDULE INITIAL PROVISIONING SHIPMENTS FOR
      C                 IMMEDIATE DELIVERY FROM THE VENDOR
        120 CONTINUE
              IDELIV=ITIME+1
              CALL ORDERV(N,IRO,IDELIV)
       2000 CONTINUE
            RETURN
            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.

```
@ 04  10-20-78   10.534   S  SPECIAL STATISTICS ROUTINE
```

```
      $RUN*;RIME/OBJ/SSTAT,O(SCD,NOGO)
      $SSTAT,S  SPECIAL STATISTICS ROUTINE
            SUBROUTINE SSTAT(IP3)
      C                 SUBROUTINE SSTAT
            COMMON/IDSTAT/IDSTAT
            COMMON/ISTAT/ISTAT
            COMMON/ISTOP/ISTOP
            COMMON/ITIME/ITIME
            COMMON/ITINV/ITINV
            COMMON/ITQTR/ITQTR
            COMMON/NITEM/NITEM
             COMMON/NBMAX/NBMAX
             COMMON/NLOCBK/NLOCBK
            COMMON/INVACT/INVACT(1)
            COMMON/INVDUE/INVDUE(1)
           COMMON/INWIP/INWIP(1)
            COMMON/NDENT/NDENT(1)
            COMMON/IBACKI/IBACKI(16,3,6)
            COMMON/IBACKT/IBACKT(16,3,6)
            COMMON/IBAKDT/IBAKDT(16,3,6)
            COMMON/IBAKDI/IBAKDI(16,3,6)
            COMMON/INVDAY/INVDAY(16,3,6)
            COMMON/INVOH/INVOH(16,3,6)
            COMMON/INVOR/INVOR(16,3,6)
           COMMON/INWIP/INWIP(16,3,6)
            COMMON/NBOTU/NBOTU(1)
             COMMON/NBOTR/NBOTR(1)
             COMMON/NBOIR/NBOIR(1)
             COMMON/NBOIU/NBOIU(1)
            COMMON/UCOST/UCOST(1)
      C
            I = ITINV
             KEND=IP3
      C
           DO 50 NN=1,NITEM
           N=NN
           KK=KNSKU(N)
            IF(INVACT(N).LE.0)GO TO 20
      C
      C                 UPDATE ON-HAND INVENTORY-WEEKS COUNTERS
      C
             CALL CUM(INVDAY,INVACT(N),N)
        20       CONTINUE
      C
      C                 IF THERE ARE NO BACKORDERS FOR ITEM N,
      C                       GO TO 50
      C
            IF(NBOTU(N).LE.0) GO TO 50
      C
      C                 UPDATE BACKORDER STATISTICS
      C
            IBAKDT(I,1,KK)=IBAKDT(I,1,KK)+NBOTR(N)
```

SSTAT

```
      CALL CUMB(IBAKDT,NBOTU(N),N)
      IF(NBOIU(N).LE.0) GO TO 50
      IBAKDI(I,1,KK)=IBAKDI(I,1,KK)+NBOIR(N)
      CALL CUMB(IBAKDI,NBOIU(N),N)
   50 CONTINUE
   60 CONTINUE
C
      KD = ITOTR/IDSTAT
      IF(KEND.GE.KD)GO TO 100
      KEND = KEND + 1
      GO TO 200
C
C                  UPDATE END OF PERIOD STATISTICS
C
  100 CONTINUE
      DO 150 NN=1,NITEM
      N=NN
      KK=KNSKU(N)
      IF(INWIP(N).GT.0)CALL CUM(IWIP,INWIP(N),N)
      IF(INVACT(N).GT.0) CALL CUM(INVOH,INVACT(N),N)
      IF(INVDUE(N).GT.0) CALL CUM(INVOR,INVDUE(N),N)
C
C                  UPDATE EOP BACKORDER STATISTICS
C
      IF(NBOTU(N).LE.0) GO TO 150
      IBACKT(I,1,KK)=IBACKT(I,1,KK)+NBOTR(N)
      CALL CUMB(IBACKT,NBOTU(N),N)
      IF(NBOIU(N).LE.0) GO TO 150
      IBACKI(I,1,KK)=IBACKI(I,1,KK)+NBOIR(N)
      CALL CUMB(IBACKI,NBOIU(N),N)
  150 CONTINUE
  160 CONTINUE
C
C                  INCREMENT PERIOD COUNTERS
C
      KEND=1
      ITINV=ITINV+1
C
C                  CREATE NEXT STATISTICS COLLECTION EVENT
C
  200 CONTINUE
      ISTAT = ISTAT + IDSTAT
      CALL ENTER(ISTAT,11,KEND,KEND,KEND)
      RETURN
      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    10.571

  1             SUBROUTINE STATHD(J,ITOTL)
  2       C      PRINT STATISTIC MEASURE FOR PARTICULAR REPORT(J=1,2,OR 3)
  3             DIMENSION ITOTL(1)
  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 40
 14            20 WRITE(6,106)
 15              GO TO 40
 16            30 WRITE(6,107)
 17           104 FORMAT(//,T56,"***   ACTIONS/MSN A   ***")
 18           107 FORMAT(//,T56,"$$$   DOLLARS   $$$")
 19           106 FORMAT(//,T56,"...   UNITS   ...")
 20            40 CONTINUE
 21              RETURN
 22              END
```

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.

```
      SUBROUTINE WRIFEL
      COMMON/NFIRST/NFIRST
      COMMON/NENTRY/NENTRY
      COMMON/JPOINT/JPOINT(1)
      COMMON/JFSN/JFSN(1)
      COMMON/JQTY/JQTY(1)
      COMMON/JPRIOR/JPRIOR(1)
      COMMON/JTYPE/JTYPE(1)
      COMMON/JTIME/JTIME(1)
      COMMON/ILOCFE/ILOCFE(1)
      WRITE(6,23)JTIME(NFIRST),NFIRST,NENTRY
   23 FORMAT(//30X,"FUTURE EVENTS LIST AT TIME ",I10/
     &       40X,"NFIRST=",I10,"     NENTRY=",I10//
     &       T20,"K     JTIME     JTYPE     JFSN     JQTY     JPRIOR",
     &       "     JPOINT     INDEX"//)
C
      K=0
      KK=NFIRST
   10 CONTINUE
      IF(KK.LE.0) GO TO 900
      K=K+1
      WRITE(6,43)K,JTIME(KK),JTYPE(KK),JFSN(KK),JQTY(KK),
     &       JPRIOR(KK),JPOINT(KK),KK
   43 FORMAT(T10,8I10)
      KK=JPOINT(KK)
      GO TO 10
  900 CONTINUE
      WRITE(6,903)
  903 FORMAT(//T30,"NO MORE ENTRIES ON THE FUTURE EVENTS LIST")
      RETURN
      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.

```
@RUN=;RIMS/QBE/ZERO.O,W(BCD,NOUOT)
*ZERO,S   ZERO STATISTICAL ACCUMULATORS
      SUBROUTINE ZERO
      COMMON/IOBLIG/IOBLIG
      COMMON/IBOP/IBOPON(3),IBOPOE(3)
      COMMON/IBAUTH/IBAUTH(12)
      COMMON/IBPROJ/IBPROJ(12)
      COMMON/IBACKI/IBACKI(16,3,6)
      COMMON/IBACKT/IBACKT(16,3,6)
      COMMON/IBAKDI/IBAKDI(16,3,6)
      COMMON/IBAKDT/IBAKDT(16,3,6)
     COMMON/IBODAT/IBODAT(16,3,6)
     COMMON/IBODAI/IBODAI(16,3,6)
      COMMON/ICANCL/ICANCL(16,3,6)
      COMMON/IDISPS/IDISPS(16,3,6)
      COMMON/IEXPED/IEXPED(16,3,6)
      COMMON/IFILLI/IFILLI(16,3,6)
      COMMON/IFILLT/IFILLT(16,3,6)
      COMMON/INVDAY/INVDAY(16,3,6)
      COMMON/INVON/INVON(16,3,6)
      COMMON/IYVON/IYVON(16,3,6)
      COMMON/IORDER/IORDER(16,3,6)
      COMMON/IRATON/IRATON(16,3,6)
      COMMON/IRECET/IRECET(16,3,6)
      COMMON/IREQC /IREQC(16,3,6)
      COMMON/IREQI /IREQI(16,3,6)
      COMMON/IREQT/IREQT(16,3,6)
      COMMON/IRETRN/IRETRN(16,3,6)
      COMMON/ISHIPI/ISHIPI(16,3,6)
      COMMON/ISHIPT/ISHIPT(16,3,6)
      COMMON/ITERM /ITERM(16,3,6)
      COMMON/ISHORD/ISHORD(16,3,6)
      COMMON/ILGORD/ILGORD(16,3,6)
     COMMON/IREPGN/IREPGN(16,3,6)
     COMMON/INRTS/INRTS(16,3,6)
     COMMON/ICUDEM/ICUDEM(16,3,6)
     COMMON/IRECPL/IRECPL(16,3,6)
     COMMON/IWIP/IWIP(16,3,6)
     COMMON/IWFP/IWFP(16,3,6)
C
      COMMON/IBOPCS/IBOPSM(3),IBOPLB(3)
C       ZERO STATISTICAL ACCUMULATORS
      IOBLIG=0
     DO 25 I=1,16
      DO 25 J=1,3
     DO 25 K=1,6
      ICANCL(I,J,K)=0
      INVON(I,J,K)=0
      IYVON(I,J,K)=0
      IRECET(I,J,K)=0
      IRETRN(I,J,K)=0
      INVDAY(I,J,K)=0
```

```
01  10-20-79   13.561      ZERO STATISTICAL ACCUMULATORS
 3              IORDER(I,J,K)=0
 4              IDISP$(I,J,K)=0
 5              ITERM(I,J,K)=0
 6              IEXPED(I,J,K)=0
 7              IRATON(I,J,K)=0
 8              IRBOC(I,J,K)=0
 9              IRBOT(I,J,K)=0
10              IRBOI(I,J,K)=0
11              IBACKT(I,J,K)=0
12              IBACKI(I,J,K)=0
13              IBAKOT(I,J,K)=0
14              IBAKOI(I,J,K)=0
15              IBODAT(I,J,K)=0
16              IBODAI(I,J,K)=0
17              IFILLT(I,J,K)=0
18              IFILLI(I,J,K)=0
19              ISHIPT(I,J,K)=0
20              ISHIPI(I,J,K)=0
21              ISHORD(I,J,K)=0
22              ILOORD(I,J,K)=0
23              IBBPOR(I,J,K)=0
24              IRETS(I,J,K)=0
25              ICRDER(I,J,K)=0
26              IRECPT(I,J,K)=0
27              IWIP(I,J,K)=0
28              IWIP$(I,J,K)=0
29          25 CONTINUE
30        C
31        C
32        C         ZERO ON-HAND AND ON-ORDER COUNTERS
33        C
34              DO 110 I=1,3
35              IBOPOH(I)=0
36              IBOPOR(I)=0
37              IBOPBK(I)=0
38              IBOPLO(I)=0
39          110 CONTINUE
40              RETURN
41              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.