

RCSL: 43-1111
AUTHOR: RES
EDITED: 77.05.30

PROGRAM RC36-90170.00
DUMP - RELEASE 2

KEYWORDS: MUSIL, CONVERSION, DPO, MTA, PRINTOUT, DATA ENTRY, LISTING

ABSTRACT: THIS PROGRAM HANDLES DATA BATCHES OR JOBS OF DATA BATCHES FROM DISC WITH A MAXIMUM RECORDSIZE OF 80 BYTES WITH ASCII CODE DATA.
OUTPUT ON NO LABELLED TAPE WITH A BLOCK SIZE OF 400 BYTES, EACH BLOCK CONSISTING OF 5 RECORDS OF 80 BYTES, IN EBCDIC CODE.
IF A RECORD IS SMALLER THAN 80 BYTES IT IS FILLED UP WITH SPACES. IF IT IS GREATER THAN 80 BYTES IT IS CUT OFF AND THE TEXT 'RECORD <NO> TOO LONG' IS WRITTEN IN THE PRINTOUT. INVALID RECORDS ARE NOT DUMPED.
NUMBER OF DUMPED RECORDS IS WRITTEN IN THE PRINTOUT.
THIS PROGRAM IS A DATA ENTRY SUPERVISOR PROGRAM.

RCSL: 43-RI0968: ASCII SOURCE TAPE
RCSL: 43-RI0969: REL. BIN TAPE
!

TITLE:

DUMP - RELEASE 2

ABSTRACT:

THIS PROGRAM HANDLES DATA BATCHES OR JOBS OF DATA BATCHES FROM DISC WITH A MAXIMUM RECORDSIZE OF 80 BYTES WITH ASCII CODE DATA.
OUTPUT ON NO LABELLED TAPE WITH A BLOCK SIZE OF 400 BYTES, EACH BLOCK CONSISTING OF 5 RECORDS OF 80 BYTES, IN ERCDIC CODE.
IF A RECORD IS GREATER THAN 80 BYTES IT IS CUT OFF AND THE TEXT 'RECORD <NO> TOO LONG' IS WRITTEN IN THE PRINTOUT. IF RECORD IS SMALLER THAN 80 BYTES IT IS FILLED UP WITH SPACE.
INVALID RECORDS ARE NOT DUMPED.
NUMBER OF DUMPED RECORDS IS WRITTEN IN THE PRINTOUT.
THIS PROGRAM IS A DATA ENTRY SUPERVISOR PROGRAM.

SIZE:

7438 BYTES. INCLUDING ONE 512 BYTES INPUT BUFFER, ONE 400 BYTES OUTPUT BUFFER AND ONE PRINTOUT-BUFFER.

DATE:

MARCH 30TH 1977.

CALL:

DUMP BATCH/JOB <BATCHNAME>/<JOBNAME> NEW/OLD
BATCH/JOB: INDICATES IF A BATCH OR A JOB IS TO BE DUMPED.
<BATCHNAME>/<JOBNAME>: THE NAME OF THE BATCH OR THE JOB THAT IS TO BE DUMPED.
NEW/OLD: NEW STARTS DUMPING AT THE BEGINNING OF THE TAPE.
 OLD STARTS DUMPING AFTER THE LAST DATABLOCK.
NEW/OLD CAN BE FOLLOWED BY DUMPOK AND/OR RELEASE:
DUMPOK: THE BATCH/JOB HAS BEEN DUMPED ONCE BEFORE, AND MUST BE DUMPED ONCE MORE.
RELEASE: THE BATCH/JOB MUST BE REKEYED AND IS NOT REKEYED YET.

OUTPUT MESSAGES:

NOT NAME NO BATCH/JOB WITH THE SPECIFIED NAME EXISTS.
NOT BATCH THE SPECIFIED BATCHNAME IS NOT A BATCH WITHIN THE SYSTEM.
SYNTAX SYNTAX ERROR IN THE CALL LINE.
CF LIST SOME ERRORMESSAGES ARE WRITTEN IN THE PRINTOUT.
 CONFER THE LOG.
BATCH IN USE THE BATCH IS USED BY ANOTHER KEYSTATION.
JOB IN USE THE JOB IS USED BY ANOTHER KEYSTATION.
LIBRARY IN USE THE JOB LIBRARY IS USED BY ANOTHER KEYSTATION.
NOT JOB THE SPECIFIED NAME IS NOT A NAME OF A JOB.
OK PROGRAM EXECUTION IS TERMINATED SUCCESSFULLY.
MAGTAPE ERROR <CODE> CONSULT THE RC3600 DATA ENTRY USER'S GUIDE, PART 2, APPENDIX 3.
DISC ERROR <CODE> CONSULT THE RC3600 DATA ENTRY USER'S GUIDE, PART 2, APPENDIX 2.
PRINTOUT ERROR <CODE> CONSULT THE RC3600 DATA ENTRY USER'S GUIDE, PART 2, SECTION 8.2.

SPECIAL REQUIREMENTS:

CMMD (R0001: RCSL: 43-RI0398).
GTPM (R0003: RCSL: 43-RI0654).
RETUR (R0004: RCSL: 43-RI0528).
ACCES (R0013: RCSL: 43-RI0976).
CONNEX (R0086: RCSL: 43-GL3275).
FITEM (R0017: RCSL: 43-RI0669)
GETNEX (R0018: RCSL: 43-RI0931)

CONST

NEW= 'NEW',
OLD= 'OLD',

RECLGT= 80, ! CONSTANT RECORD LENGTH !

FF= '<12>',
JBLIB= 'JBLIR<0>',
TBATCH= 'BATCH',
JOB= 'JOB',
HJOB= 'JOB',
TXNLSP= '<13><10>',
TXSPACES= '<13><10>NO OF DUMPED RECORDS:',
SPACES= ' ',
EM= '<25>',
NULL= '<0>',
NOTBATCH= ': NOT BATCH<13><10>',
TXEX= ': DOES NOT EXIST<13><10>',
STATERR= ': NOT DUMPED, STATE ERROR<13><10>',
BUSED= ': NOT DUMPED, IN USE<13><10>',
NOREK= 'RELEA',
DMPOK= 'DUMPO',
HEADLINE= '<13><10>* * * * * * * * * * * DUMP OF * * * * *',
TEXTLINE= '<13><10>BATCHNAME FORMAT NUMBER OF DUMPED RECORDS<13><10>',
TXTOTAL= '<13><10>THE TOTAL NUMBER OF DUMPED RECORDS:',
TXEND= '<13><10>* * * * * * * * * * * END OF DUMP * * * * *',
TXLONG= '<13><10>RECORD NUMBER TOO LONG',
TXINVAL= '<13><10>RECORD NUMBER INVALID, NOT DUMPED',
SP= '<32>',
NL= '<13><10>',

ASDICTAB= ! ASCII TO EBCDIC CONVERSION TABLE
0 1 2 3 4 5 6 7 !

#	!	0	!	0	1	2	3	55	45	46	47
!	8	!	22	5	37	11	12	13	14	15	
!	16	!	16	17	18	19	60	61	50	38	
!	24	!	24	25	63	39	28	29	30	31	
!	32	!	64	90	127	123	91	108	80	125	
!	40	!	77	93	92	78	107	96	75	97	
!	48	!	240	241	242	243	244	245	246	247	
!	56	!	248	249	122	94	76	126	110	111	
!	64	!	124	193	194	195	196	197	198	199	
!	72	!	200	201	209	210	211	212	213	214	
!	80	!	215	216	217	226	227	228	229	230	
!	88	!	231	232	233	123	124	91	109	63	
!	96	!	63	129	130	131	132	133	134	135	
!	104	!	136	137	145	146	147	148	149	150	
!	112	!	151	152	153	162	163	164	165	166	
!	120	!	167	168	169	63	63	208	95	7	
#;											

```

COMMAND LINE
PARAMETER ADDR OF MESSAGE BUF
PARAMETER TEXT
PARAMETER VALUE
PARAMETER KIND
PARAMETER SEPERATOR

BLOCK: INTEGER; ! NUMBERS OF BLOCKS ON OLD TAPE
FIELD: STRING(80); ! FIELD IN RECORD
SUBNAME: STRING(2); ! SUBNAME OF SUBFORMAT OF A RECORD

X: INTEGER; ! NUMBER OF CHAR IN BUFFER
Y: INTEGER; ! NUMBER OF CHAR IN FIELD
V: INTEGER;
LASTPOS: INTEGER;

COM: STRING(5); ! COMMAND JOB OR BATCH
JOBNAME: STRING(6); ! JOBNAME OR BATCHNAME
EWORD: INTEGER; ! JOBNAME, EWORD AND RES
RES: INTEGER; ! MUST BE CONSECUTIVE
CTYPE: INTEGER;
LENGTH: INTEGER;
SUM: INTEGER; ! TOTAL DUMPED RECORDS
STATUS: INTEGER; ! BATCH STATUS
TLSWITCH: INTEGER; ! BOOLEAN FOR RECORD TOO LONG
REC: STRING(80); ! WORKING AREA FOR CURR.RECORD
DUMPNO: INTEGER; ! NUMBER OF DUMPED RECORDS
ITEM1: STRING(5);
ITEM2: STRING(6);
FLAG: INTEGER;
RELEASE: INTEGER;
NOHELP: STRING(2);
NO: INTEGER; ! NUMBER OF CHAR IN REC
RECNO: INTEGER; ! CURR. RECORD NUMBER
BINNO: INTEGER; ! WORKING AREA
NAME: STRING(6); ! BATCH NAME
DECNO: STRING(5); ! WORKING LOCATION
ERRLPT: INTEGER; ! NUMBER OF ERRORS ON LPT
ERRBAT: INTEGER; ! NUMBER OF ERRORS ON BATCH
ERRJAT: INTEGER; ! NUMBER OF ERRORS ON JOB
ERRMTO: INTEGER; ! NUMBER OF ERRORS ON MTO

RTEXT1: INTEGER; ! RETURN PARAMETER TEXT1
RTEXT2: INTEGER; ! RETURN PARAMETER TEXT2
RSPEC: INTEGER; ! RETURN PARAMETER SPECIAL
RCON: INTEGER; ! RETURN PARAMETER CONTINUE
RESULT: INTEGER; ! RETURN PARAMETER RESULT
TEXTMODE: INTEGER; ! RETURN PARAMETER MODE
LNAME: STRING(6); ! NAME OF PRINTEROUTPUT-FILE

```

```

LPT:      FILE          ! PRINTER FILE DESCRIPTION
          'LPT',        ! NAME OF PRINTER DRIVER
          62,           ! KIND = CHARACTER ORIENTED
          1,            ! BUFFERS
          512,         ! SHARESIZE
          U;           ! FORMAT = UNDEFINED BLOCKED

```

```

GIVEUP
LPERROR,  ! BATCH ERROR PROCEDURE
2'1111111111111111

```

```

OF STRING(512);  ! RECORD STRUCTURE

```

```

BATCHNAME: STRING(2);  ! CHANGE BATCH NAME

```

```

BATCH:    FILE          ! BATCH ERROR DESCRIPTION
          'BATCH',      ! NAME OF BATCH
          60,           ! KIND = POSITIONABLE, REPEATABLE
          1,            ! BUFFERS
          512,         ! SHARESIZE
          U;           ! FORMAT = UNDEFINED

```

```

GIVEUP
BATERROR, ! BATCH ERROR PROCEDURE
2'1111111111111111

```

```

OF RECORD      ! RECORD STRUCTURE

```

```

      JNAME: STRING(6) FROM 1;
      BNAME: STRING(6) FROM 13;
      STAT:  STRING(2) FROM 19;
      FORM:  STRING(6) FROM 7

```

```

END;

```

```

MTO:      FILE          ! KIND = REPEATABLE, POSITIONABLE,
          'MTO',        ! BLOCKED
          14,           !
          1,            ! BUFFERS
          400,         ! SHARESIZE
          FB;          ! FORMAT = FIXED BLOCKED

```

```

GIVEUP
MTERROR,  ! MT ERROR PROCEDURE
2'111110111110011111

```

```

OF STRING(80);  ! RECORD STRUCTURE

```

06.78

REGNECENTRALEN

```

PROCEDURE CMD (VAR COMLINE: STRING(112); ! GET COMMAND
              VAR PRETUR: INTEGER );
CODEBODY;

PROCEDURE GTPM (VAR COMLINE: STRING(112); ! GET PARAMETER
              VAR PTEXT: STRING(5) ;
              VAR PVALUE: INTEGER ;
              VAR PKIND: INTEGER ;
              VAR PSEP: INTEGER );
CODEBODY;

PROCEDURE RETUR (VAR PRETUR: INTEGER ; ! RETURN
               VAR RESULT: INTEGER ;
               VAR RTEXT1: INTEGER ;
               VAR TEXTMODE: INTEGER ;
               VAR COMLINE: STRING(112));
CODEBODY;

PROCEDURE ACCES (FILE BATCH ; ! GET PART OF RECORD
               VAR FIELD: STRING(80);
               VAR Y : INTEGER ;
               VAR STATUS: INTEGER ;
               VAR SUBNAME: STRING(2) );
CODEBODY;

PROCEDURE CONNEX (FILE LPT ;
                 VAR X: INTEGER ;
                 VAR LNAME: STRING(6) );
CODEBODY;

PROCEDURE FITEM (FILE BATCH;
                CONST CTYPE: INTEGER;
                CONST LENGTH: INTEGER;
                VAR NAME: STRING(6));
CODEBODY;

PROCEDURE GETNEX (FILE BATCH;
                 CONST LENGTH: INTEGER;
                 VAR NAME: STRING(6);
                 VAR EWORD: INTEGER;
                 VAR RES: INTEGER);
CODEBODY;

PROCEDURE REPOS;
BEGIN
  SETPOSITION(MTD,1,BLOCK);
  Y := 400 - LASTPOS;
  WHILE Y <> MTD.ZREM DO
    PUTREC(MTD,80);
  IF RTEXT1 = 42 THEN GOTO 45; ! THE PROCEDURE IS CALLED BECAUSE !
                              ! OF A PRINTOUT ERROR !

  DUMPNO := 0;
  BINDEC(DUMPNO,DECNO);
  OUTTEXT(LPT,DECNO);
  OUTTEXT(LPT,NL);
  IF COM = JOB THEN BEGIN
    BINDEC(SUM,DECNO);
    OUTTEXT(LPT,TXTOTAL);
    OUTTEXT(LPT,DECNO);
    OUTTEXT(LPT,NL);
  END;
  OUTTEXT(LPT,TXEND);
45:
END;

```

03.76 11.479

REGNOCENTRALEN

311 662

PROCEDURE LPTERROR;

BEGIN

```

RTEXT1 := 42;           ! PRINTOUT
RTEXT2 := 7;           ! ERROR
RSPEC := 1;
RCON := 0;
RESULT := LPT.70;
ERRLPT := ERRLPT + 1;
REPOS;
FLAG := 0;
GOTO 200;

```

END;

PROCEDURE OUTNAME;

BEGIN

```

V := 0;
WHILE V < 5 DO BEGIN
  MOVE(ITEM1,V,DECNO,0,1);
  IF DECNO = NULL THEN MOVE(SP,0,DECNO,0,1);
  MOVE(DECNO,0,ITEM2,V,1);
  V := V + 1;

```

END;

MOVE(NULL,0,ITEM2,5,1);

END;

PROCEDURE CHECKSTAT;

BEGIN

```

IF STATUS AND 8'174000 = 8'100000 THEN GOTO 60; ! CLOSED !
IF STATUS = 0 THEN GOTO 63; ! EMPTY !

```

GOTO 62;

60: IF RELEASE = 3 THEN GOTO 63; ! NO FURTHER CHECK !

IF RELEASE = 2 THEN GOTO 61;

```

IF STATUS AND 8'3000 = 8'2000 THEN GOTO 62; ! MUST BE REKEYED !
! AND REKEYED !

```

IF STATUS AND 2 = 2 THEN

```

IF STATUS AND 8'10 <> 0 THEN GOTO 62; ! VALID REQUIRED !
! AND NOT INVALID !

```

61: IF RELEASE = 1 THEN GOTO 63;

IF STATUS AND 8'40 = 8'40 THEN GOTO 62; ! DUMPED !

GOTO 63;

62: OUTTEXT(LPT,STATERR); ! STATE ERROR !

STATUS := 0; ! NOT DUMPED !

RTEXT1 := 13;

GOTO 64;

63: STATUS := 1;

64:

END;

PROCEDURE DUMPBIT;

BEGIN

SETPOSITION(BATCH,0,0);

GETREC(BATCH,NO);

STATUS := WORD BATCH↑.STAT;

IF STATUS AND 8'40 = 0 THEN BEGIN

NO := STATUS + 8'40;

CLOSE(BATCH,1);

OPEN(BATCH,3);

SETPOSITION(BATCH,0,0);

BATCH.ZFIRST := BATCH.ZTOP;

MOVE(NOHELP,2,BATCH↑.18,2);

BATCH.ZREM := 0;

OUTBLOCK(BATCH);

END;

END;

0375 3160

PROCEDURE BATERA
1000

```

IF CON = BATCH THEN BEGIN
  RTEXT1 := 5;      ! BATCH !
55:  RTEXT2 := 43;   ! IN USE!
  RSPEC := 0;
  RCON := 0;
  RESULT := 0;
  FLAG := 0;
  GOTO 200;
END;
IF BATCH.ZNAME = JOBNAME THEN BEGIN
  RTEXT1 := 45;
  GOTO 55;
END;
OUTTEXT(LPT,BUSED);
RTEXT1 := 13;
RES := 2;
GOTO 70;
END;

```

PROCEDURE BATEROR;
BEGIN

```

  RSPEC := 1;
  RCON := 0;
  RESULT := BATCH.Z0;
  IF BATCH.Z0 AND 8'1000 <> 0 THEN BATCHSTATE;
  IF BATCH.Z0 AND 8'040000 <> 0 THEN
    BEGIN
      IF FLAG = 0 THEN BEGIN
        RTEXT1 := 11;          ! NOT NAME !
        RTEXT2 := 0;
        RSPEC := 0;
        GOTO 10;
      END;
      OUTTEXT(LPT,TEX);
      RTEXT1 := 13;
      RES := 2;
      GOTO 70;
    END;
  END;
  RTEXT1 := 15;          ! DISC !
  RTEXT2 := 7;          ! ERROR !
  REPOS;
10: IF BATCH.ZNAME <> JOBNAME THEN ERBBAT := ERBBAT + 1;
    IF BATCH.ZNAME = JOBNAME THEN ERRJAT := ERRJAT + 1;
    GOTO 200;
END;

```

NR 802
REGNOCENTRALEN

PROCEDURE MTEERROR;

BEGIN

IF MTO.ZO AND 8'10 <> 0 THEN GOTO 40;

IF MTO.ZO AND 8'400 = 0 THEN BEGIN

RTEXT1 := 8;

! MAGTAPE !

RTEXT2 := 7;

! ERROR !

RSPEC := 1;

RCON := 0;

RESULT := MTO.ZO;

ERRMTO := ERRMTO + 1;

! MT-ERROR: NOISE RECORD, BLOCK LENGTH, DATA CHANNEL,
PARITY, EOT !

IF MTO.ZO AND 8'10360 <> 0 THEN REPOS;

GOTO 200;

END;

40:

END;

PROCEDURE SYNTAX;

! PARAMETER ERROR !

BEGIN

RTEXT1 := 4;

! SYNTAX !

RTEXT2 := 0;

RSPEC := 0;

RCON := 0;

RESULT := 0;

GOTO 300;

END;

PROCEDURE NOTNAME;

BEGIN

IF COM = JOB THEN IF BATCH.ZNAME = NAME THEN BEGIN

OUTTEXT(LPT,NOTBATCH);

CLOSE(BATCH,1); ERBBAT := 4;

GOTO 70;

END;

RTEXT1 := 24;

! NOT !

IF COM = JOB THEN RTEXT2 := 45;

! JOB !

IF COM = TBATCH THEN RTEXT2 := 5;

! BATCH !

RSPEC := 0;

RCON := 0;

RESULT := 0;

GOTO 200;

END;

PROCEDURE WRITEREC;

BEGIN

IF NO <> 0 THEN

BEGIN

IF NO > RECLGT THEN

BEGIN

ILSWITCH := 1;

BINDEC(RECNO,DECNO);

MOVE(DECNO,0,TXLONG,16,5);

OUTTEXT(LPT,TXLONG);

RTEXT1 := 13;

END;

WHILE NO < RECLGT DO

BEGIN

MOVE(SP,0,REC,NO,1);

NO := NO + 1;

END;

DUMPNO := DUMPNO + 1;

PUTREC(MTO,RECLGT);

CONVERT(REC,MTO↑,ASDICTAB,RECLGT);

NO := 0;

END;

0676
CENTRALEN

PROCEDURE DUMPRATCH;

BEGIN

```

OUTTEXT(LPT, TXNLSP);
ITEM1 := NAME; OUTNAME; OUTTEXT(LPT, ITEM2);
BATCH.ZNAME := NAME;
OPEN(BATCH, 1); ERBBAT := 0; ! OPEN DISC FOR READING
RECNO := 1; NO := 0; DUMPNO := 0;
SETPOSITION(BATCH, 0, 0);
GETREC(BATCH, X); ! GET BATCHHEAD
IF BATCH.ZNAME <> BATCH↑.RNAME THEN NOTNAME;
IF BATCH↑.JNAME <> JOBNAME THEN NOTNAME;
STATUS := WORD(BATCH↑.STAT);
CHECKSTAT;
IF STATUS = 0 THEN GOTO 30;
OUTTEXT(LPT, SPACES);
ITEM1 := BATCH↑.FORM; OUTNAME; OUTTEXT(LPT, ITEM2);
OUTTEXT(LPT, SPACES);
TLSWITCH := 0;
SETPOSITION(BATCH, 0, 1);
5: ACCES(BATCH, FIELD, Y, STATUS, SUBNAME);
IF Y >= 0 THEN BEGIN ! A FIELD IS READ !
  V := Y;
  IF NO + V > RECLGT THEN BEGIN
    V := 80 - NO;
    IF V < 0 THEN V := 0;
  END;
  MOVE(FIELD, 0, REC, NO, V);
  NO := NO + Y;
  GOTO 10;
END;
IF Y = -1 THEN BEGIN ! A RECORD HEAD IS READ !
  IF STATUS <> 0 THEN BEGIN
    TLSWITCH := 1;
    BINDEC(RECNO, DECNO);
    MOVE(DECNO, 0, TXINVAL, 16, 5);
    OUTTEXT(LPT, TXINVAL);
    REPEAT ACCES(BATCH, FIELD, Y, STATUS, SUBNAME)
    UNTIL Y = -2;
    RECNO := RECNO + 1;
  END;
  GOTO 10;
END;
IF Y = -2 THEN BEGIN ! A RECORD END IS READ !
  WRITEREC;
  RECNO := RECNO + 1;
  GOTO 10;
END;
IF Y = -3 THEN GOTO 20; ! A BATCH END IS READ !
10: GOTO 5;
20: IF TLSWITCH = 1 THEN BEGIN
  OUTTEXT(LPT, TXSPACES);
  END;
  BINDEC(DUMPNO, DECNO);
  OUTTEXT(LPT, DECNO); OUTTEXT(LPT, NL);
  BLOCK := MTO.ZBLOCK; LASTPOS := 400 - MTO.ZREM;
  DUMPRIT;
30: CLOSE(BATCH, 1); ERBBAT := 4;
END;

```

06.76 31489

REGNECENTRALEN

1.862

PROCEDURE CHECKKIND;

BEGIN

```

IF COM = TRATCH THEN BEGIN
  BATCH.ZNAME := NAME;
  OPEN(BATCH,1);  ERRBAT := 0;
  SETPOSITION(BATCH,0,0);  GETREC(BATCH,X);
  IF BATCH.ZNAME <> BATCH↑.BNAME THEN NOTNAME;
  MOVE(BATCH↑,0,JOBNAME,0,6);
  CLOSE(BATCH,1);  ERRBAT := 4;

```

END;

```

BATCH.ZNAME := JRLIB;
OPEN(BATCH,1);  ERRBAT := 0;
SETPOSITION(BATCH,0,0);
LENGTH := 3;  CTYPE := 0;
FITEM(BATCH,CTYPE,LENGTH,JOBNAME);
IF RES = 1 THEN NOTNAME;
CLOSE(BATCH,1);  ERRBAT := 4;

```

END;

0676

RIEGNECENTRALEN

111 892

BEGIN

CMMD(COMLINE,PTEXT);

! GET COMMAND

ERRJAT := 4; ERRPNT0 := 4; ERRDAT := 4; ERRPRT := 4;

GTPM(COMLINE,PTEXT,PVALUE,PKIND,PSEP); ! GET PARAMETER - MUST BE DUMP

LNAME := PTEXT;

FLAG := 0; RELEASE := 0;

GTPM(COMLINE,PTEXT,PVALUE,PKIND,PSEP); ! GET PARAMETER - MUST BE
! JOB / BATCH

IF PKIND <> 1 THEN SYNTAX;

! ELSE SYNTAX

IF PSEP = 2 THEN SYNTAX;

COM := PTEXT;

IF COM <> TBATCH THEN IF COM <> JOB THEN SYNTAX;

GTPM(COMLINE,PTEXT,PVALUE,PKIND,PSEP); ! GET PARAMETER - MUST BE
! JOB- OR BATCHNAME

IF PKIND <> 1 THEN SYNTAX;

IF PSEP = 2 THEN SYNTAX;

IF COM = TBATCH THEN NAME := PTEXT;

IF COM = JOB THEN JOBNAME := PTEXT;

BATCH.ZNAME := PTEXT;

OPEN(BATCH,1); ERBAT := 0;

CLOSE(BATCH,1); ERBAT := 4;

GTPM(COMLINE,PTEXT,PVALUE,PKIND,PSEP); ! GET PARAMETER OLD/NEW

IF PKIND <> 1 THEN SYNTAX;

ITEM2 := PTEXT;

GTPM(COMLINE,PTEXT,PVALUE,PKIND,PSEP);

IF PKIND = 2 THEN GOTO 40;

IF PTEXT <> DMPOK THEN IF PTEXT <> NOREK THEN SYNTAX;

IF PTEXT = DMPOK THEN RELEASE := 1;

IF PTEXT = NOREK THEN RELEASE := 2;

GTPM(COMLINE,PTEXT,PVALUE,PKIND,PSEP);

IF PKIND = 2 THEN GOTO 40;

IF PKIND <> 1 THEN SYNTAX;

IF PTEXT <> DMPOK THEN IF PTEXT <> NOREK THEN SYNTAX;

IF PTEXT = DMPOK THEN RELEASE := RELEASE + 1;

IF PTEXT = NOREK THEN RELEASE := RELEASE + 2;

IF RELEASE > 3 THEN SYNTAX;

IF RELEASE = 2 THEN IF PTEXT = DMPOK THEN SYNTAX;

40:

IF PSEP <> 2 THEN SYNTAX;

CHECKKIND;

RTEXT1 := 0;

IF ITEM2 = NEW THEN

BEGIN

OPEN(MT0,3); SETPOSITION(MT0,1,1);

ERRPNT0 := 0;

BLOCK := 1;

LASTPOS := 0;

GOTO 50;

END;

! OPEN MAGTAPE FOR READING

```

IF ILEN2 <> OLD THEN SYNTAX;
OPEN(MTO,5);
ERRMTO := 0;
SETPOSITION(MTO,1,32000);
BLOCK := MTO.ZBLOCK - 1;
SETPOSITION(MTO,1,BLOCK);
INBLOCK(MTO);
Y := MTO.ZREM;
LASTPOS := MTO.ZREM;
CLOSE(MTO,0);
OPEN(MTO,3);
SETPOSITION(MTO,1,BLOCK);
Y := 400 - Y;
WHILE Y <> MTO.ZREM DO
  PUTREC(MTO,80);

```

50:

```

X := 3;
CONNEX(LPT,X,LNAME); ERRLPT := 0;
OUTTEXT(LPT,FF);
FLAG := 1;
IF COM = JOB THEN
  BEGIN
    MOVE(HJOB,0,HEADLINE,33,5);
    MOVE(JOBNAME,0,ITEM1,0,5);
  END;
IF COM = TBATCH THEN
  BEGIN
    MOVE(TBATCH,0,HEADLINE,33,5);
    MOVE(NAME,0,ITEM1,0,5);
  END;
OUTNAME;
MOVE(ITEM2,0,HEADLINE,39,5);
OUTTEXT(LPT,HEADLINE);
OUTTEXT(LPT,TEXTLINE);
IF COM = TRATCH THEN
  BEGIN
    DUMPBATCH;
    GOTO 100;
  END;
BATCH.ZNAME := JORNAME;
OPEN(BATCH,1); ERRJAT := 0;
SETPOSITION(BATCH,0,0);
LENGTH := 4;
RES := 0; SUM := 0;
WHILE RES <> 1 DO BEGIN
  IF RES = 2 THEN RES := 1;
  BATCH.ZNAME := JORNAME;
  GETNEX(BATCH,LENGTH,NAME,WORD,RES);
  IF RES = 1 THEN GOTO 70;
  DUMPBATCH;
  SUM := SUM + DUMPNO;

```

70:

```

END;
CLOSE(BATCH,1); ERRJAT := 4;
BINDEC(SUM,DECNO);
OUTTEXT(LPT,TXTOTAL);
OUTTEXT(LPT,DECNO);
OUTTEXT(LPT,NL);

```

0676

GENECENTRALEN

100:

```
OUTTEXT(LPT,IXEND);
RSPEC := 0; ! NORMALLY RETURN
IF RTEXT1 = 0 THEN RTEXT1 := 12;
RTEXT2 := 0;
RCOM := 0;
RESULT := 0;
```

200:

```
WHILE ERRMTO < 4 DO
  BEGIN CLOSE(MTO,1); ERRMTO := 4 END;
WHILE ERBAT < 4 DO
  BEGIN CLOSE(BATCH,1); ERBAT := 4 END;
WHILE ERRJAT < 4 DO BEGIN
  BATCH.ZNAME := JOBNAME; CLOSE(BATCH,1); ERRJAT := 4; END;
WHILE ERRLPT < 4 DO
  BEGIN
    IF FLAG = 1 THEN OUTTEXT(LPT,EM);
    CLOSE(LPT,1); ERRLPT := 4 END;
```

300:

```
RTEXT1 := RTEXT1 SHIFT 10 + RTEXT2 SHIFT 4
        + RSPEC SHIFT 1 + RCOM ;
IF FLAG = 1 THEN RTEXT1 := RTEXT1 + 2 SHIFT 1;
TEXTMODE := 0;
RETUR(PRETUR,RESULT,RTEXT1,TEXTMODE,COMLINE); ! RETURN
END;
```