CYBERDATA INTERNAL

MAINTENANCE SPECIFICATION

22266900

INDEX

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME ___AID___   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   AID controls tutorial mode through bit 4 in Switch 1.

   Function Format:

       INT AID REL

   Both for turn on and off tutorial mode.

2. DETAILED DESCRIPTION

   The program checks format validity. In case of error bit 8 of Switch 1 is set (indicates that RESET must regenerate display) and exit is made via ERROR.

   When the format is correct, bit 4 of Switch 1 is complemented (turn on or off tutorial mode) and DISPLAY is invoked. Exit is made via CLRSR.

INTERFACE

1. ENTRY METHOD

   AID is a disk resident program loaded by INTRGT (by use of ENTPRG) and JMP is made to its single entry point.

2. EXIT METHOD

   a. JMP - (ERROR) in case of syntactic error.

   b. JMP - (CLRSR)

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. TRANSFER VECTOR

      GETCOM ($9C)
      CLRSR ($93)
      ERROR ($96)
      DISPLY ($A0)

   b. SST, ATT

      SST ($47)
      STEMP (20)
      SW1 (19)

   c. GENERAL

      ONEBIT ($23)
      ZROBIT ($33)
      ERRE1 ($12)

AA5641   PRINTED IN THE USA

2. EXTERNALS

   REGEN

3. MACROS USED

   None

4. PROGRAM DEPENDENCIES

   GETCOM
   CLRSR
   ERROR
   DISPLY

5. SYSTEM TABLES AND AREAS USED

   SST
   ATT
   IRB

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   15_{16} - 21_{10} words of program.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY

   Reentrant.

DOCUMENT CLASS __ARITH__

PROGRAM NAME____Arithmetic, conversion, and parking utilities____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   ARITH contains 8 entry points:

   ADD adds two BCD numbers.

   SUB subtracts two BCD numbers.

   DB is a decimal to binary conversion routine.

   BD is a binary to decimal conversion routine.

   PACK packs an ASCII string to a BCD number.

   UNP unpacks a BCD number to an ASCII string.

   LUL performs lower and upper limits checks on a field.

   NZT performs a nonzero test on a field.

2. DETAILED DESCRIPTION

   a. SUB

      ● Call MOVE to move the operands addressed in Q and A to locations AA through AA + 3, and BB through BB + 3, respectively.

      ● Complement bits 12-15 of BB.

      ● Move return address from SUB to ADD, and continue at step b.2.

   b. ADD

      1. (ADD). Call MOVE to move the operands addressed in Q and A to locations AA through AA + 3, and BB through BB + 3, respectively.

      2. (ADD1). Set (AA = 1) = 0 and (BB - 1) = 0 (Sign extension areas).

         ● Save address of result in ADDRR.

         ● If (AA) is negative, call TENCMP to convert number in AA through AA + 3 to 10's complement.

         ● If (BB) is negative, call TENCMP to convert the number in BB through BB + 3 to 10's complement.

      3. (ADD3). Add 16 digits in AA . . . AA + 3 to 16 digits in BB . . . BB + 3, working from right to left and storing the result in RR . . . RR + 3.

      4. (ADD9). If sign of result, in RR-1 is negative, call TENCMP to complement the result, and set bits 12-15 of RR-1 to $F.

- If sign of result is positive, set bits 12-15 of RR-1 to 0.

- If result address was specified, move result from RR . . . RR + 3 to result area.

5. (ADD11). Set (I) = Base address of ATT.

- Set (Q) = RR (Result address), and exit to caller.

c. DB

1. (DB). Save the 4-digit BCD number in SST/STEMP.

- Set SST/STEMP + 1 = -4 (counter)

2. (DBI). Convert the number to binary using the formula:

$$b = ((d_3 \times 10 + d_2) \times 10 + d_1) \times 10 + d_0$$

where b is the binary result, d_3 is the most significant digit, and d_0 is the least significant digit in the BCD number.

- Exit to the caller with (A) = (Q) = binary result

d. BD

1. (BD). Set SST/STEMP = 0   (result).

Set SST/STEMP + 1 = 4 (counter).

2. (BD1). Convert binary number in A to BCD number by dividing the binary number by 10, shifting the remainder into the BCD number from left to right; dividing the quotient by 10, shifting the remainder into the BCD number, and so forth, until a 4-digit number has been converted.

3. (BD2). Exit to the caller with (A) = BCD equivalent of binary input.

e. PACK

1. (PACK). Set PN through PN + 3 to zero (Packed number buffer).

- Adjust ATT/RCNT to point to beginning of current field.

- Set (LOPCNT) = complement of field character count.

2. (PACK1). Call GBYTE to extract next character from current field.

- Increment ATT/RCNT by 2.

- If current character is plus sign, set bits 12-15 of PN to zero and go to step 3.

- If current character is minus sign, set bits 12-15 of PN to $F and go to step 3.

DOCUMENT CLASS ___ARITH___

PROGRAM NAME___Arithmetic, conversion, and parking utilities___

- ● Convert an ASCII number to a BCD number by subtracting $30 from the ASCII number (if $20 encountered set to 0).

- ● Insert the BCD number in bits 0-3 of PN + 3, shifting all preceding digits 4 bits to the left.

- ● Increment LOPCNT by 1. Repeat step 2 until the entire field has been converted.

3. (PAKEXT). Exit to caller with (Q) = address of first word of packed result.

f. UNP

1. (UNP). Save address of packed number in UPADDR.

- ● Save bits 12-15 (sign bits) of first word in SST/STEMP.

- ● Set bits 12-15 to zero in first word of packed number.

- ● Shift packed number one digit (4 bits) to the left.

- ● Call UPASCI to convert 15 packed digits to ASCII equivalents, storing the results, two digits per word, in UPN through UPN + 7. Bits 0-7 of UPN + 7 contain $30.

- ● Set I = base address of Active Terminal Table (ATT).

- ● Restore bits 12-15 of first word of packed number to the original value.

- ● If sign is plus, set bits 0-7 of UPN + 7 to $2B (plus sign).

- ● If sign is minus, set bits 0-7 of UPN + 7 to $2D (minus sign).

- ● Exit to caller with (Q) = address of first word of ASCII result.

g. LUL

1. (LUL). Set LL through LL + 7 to zeros.

- ● Call FMTPAR to extract limit description byte from Format table. Save address of first word of lower limit value in LMTPNT. Save limit description byte in LMTTYP.

- ● Call EXPAND to move lower limit field to LL through LL + 3.

- ● Call EXPAND to move upper limit field to UL through UL + 3.

- ● Call PACK to pack the data field to be checked.

- ● Call SUB to subtract the Lower Limit value from the packed data field value, leaving the result in a data area in SUB. If Lower Limit value is greater than data field value, go to step 2.

- ● If not, call SUB to subtract the packed data field value from the Upper Limit value, leaving the result in a data area in

/53

DOCUMENT CLASS ____ARITH_____

PROGRAM NAME_____Arithmetic, conversion, and parking utilities_____

SUB. If the Upper Limit value is less than the data field value, go to step 2.

● The data field value lies within the limits. If the limits are exclusive, exit to the caller. If not, go to step 3.

2. (OUTLMT). The data field value lies outside the limits. If the limits are exclusive, exit to the caller. If not go to LULERR with (Q) = 8 (Limit Error).

3. (LULERR). If the whole field is zero, take normal exit. Otherwise, exit to ERROR with (Q) = 8 (Limit Error).

h. NZT

1. (NZT). Adjust ATT/RCNT to point to beginning of current field.

● Set SST/STEMP + 1 = complement of field character count.

● If ATT/FDTP/13-15 = 7 (signed Numeric Field), increment counter by 1, so that last character (sign) is not checked.

2. (NZTI). Call GBYTE to extract next character from current field.

● Increment ATT/RCNT by 1.

● If current character of alpha or mixed numeric field is not blank, go to step 3.

● If current character of numeric field is not zero, go to step 3.

● Increment counter in SST/STEMP + 1 by 1.

● Repeat step 2 until entire field has been checked.

● If alpha or mixed numeric field is all blank, or numeric field is all zero, restore ATT/RCNT to original value and exit to ERROR with (Q) = $C (nonzero error).

3. (NZTEXT). Restore ATT/RCNT to original value and exit to caller.

INTERFACE

1. ENTRY AND EXIT CONDITIONS

a. ADD
```
   LDA   C        Address of result; (1) =0 means result is to be
   STA-  1        left in a data area in ADD.
   LDQ   A        Address of first operand
   LDA   B        Address of second operand
   EXT   ADD
   RTJ   ADD
   --Return--
```

(Q) = address of result
(1) = address of ATT

DOCUMENT CLASS ___ARITH_____

PROGRAM NAME_____Arithmetic, conversion, and parking utilities_____

b. SUB

```
LDA    C          Address of result; (I) = 0 means result is to be
STA-   I             left in a data area in SUB.
LDQ    A          Address of first operand
LDA    B          Address of second operand
EXT    SUB
RTJ    SUB
--Return--
```

(Q) = address of result
(I) = address of ATT

c. DB

```
LDA    A          Packed BCD number in the range 0-9999
EXT    DB
RTJ    DB
--Return--
```

(A) = Binary equivalent of input number

d. BD

```
LDA    A          Binary number in the range 0-9999
EXT    BD
RTJ    BD
--Return--
```

(A) = Packed BCD equivalent of input number

e. PACK

```
EXT    PACK       Field to be packed is defined in ATT.
RTJ    PACK
--Return--
```

(Q) = Address of packed BCD number

f. UNP

```
LDA    A          Address of packed BCD number
EXT    UNP
RTJ    UNP
--Return--
```

(Q) = Address of 8-word unpacked ASCII string

g. LUL

```
EXT    LUL        Field is defined in ATT.
RTJ    LUL        Limits are defined in FMT.
--Return--
```

Return to caller if field passes limit check.
Exit to ERROR if field fails limit check.

/55

DOCUMENT CLASS __ARITH__

PROGRAM NAME__Arithmetic, conversion, and parking utilities__

h. NZT

```
EXT   NZT        Field is defined in ATT.
RTJ   NZT
--Return--
```

Return to caller if alpha field has at least one nonblank character, or if numeric field has at least one nonzero character.

Exit to ERROR otherwise.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

   ```
   LPMASK ($2)
   NZERO ($12)
   ZERO ($22)
   ONEBIT ($23)
   ZROBIT ($33)
   TEN ($46)
   SST ($47)
   GBYTE ($94)
   ERROR ($96)
   ```

   b. SST

   ```
   CATT (11)
   CFFA (12)
   STEMP (20)
   ```

   c. ATT

   ```
   FCNT (3)
   RCNT (4)
   FDTP (13)
   ```

2. MACROS USED

   None

3. PROGRAM DEPENDENCIES

   GSE   GBYTE } Get current character from Terminal Record Buffer
         ERROR } Display error code
   FMTPAR   -   Extract limit parameters

4. DATA AREAS USED

   SST   System Status Table
   ATT   Active Terminal Table

/57

DOCUMENT CLASS ARITH

PROGRAM NAME Arithmetic, conversion, and parking utilities

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$1E3_{16} = 483_{10}$ words

2. RELOCATION TYPE

ARITH is core resident, loader relocatable.

3. ENTRANCY CLASSIFICATION

Reentrant.

/56

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ BORENT - Beginning of Record (Entry Mode)   LEVEL 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   BORENT is entered when the first character (or function) of a record is depressed. It performs the following operations:

   a. Gets a new track for the batch, if required, and calculates new disk addresses.

   b. Builds TRB header for the record.

   c. Enters autosequence number when autosequencing is specified.

   The closed subroutine LINKTR included in BORENT updates the track table for the batch and writes it back to the first sector of the batch.

2. DETAILED DESCRIPTION

   BORENT calls the common entry subroutine to store the return address in the ATT exits stack. The record length is obtained from the format table and a check is made to see if sufficient room remains in the current track.

   If there is not enough room in the existing track, DAM (GITRT) is called to allocate another track. The closed subroutine LINKTR is entered to update the track table for the batch and write it back to the first sector. PIO (CLCDWA) is called to convert sector to word address and the word address is stored at ATT/NXTDA and ATT/NXTDA + 1. If the disk is full, the error is displayed at the terminal via GSE (ERROR). If this new record is an insert, and the terminal basic mode is Entry, the last entry record is read in via SUBR, an internal subroutine to Read/Write 2 Words, so that its next disk address pointers can be updated. The last entry record is then rewritten via the same routine. In Entry mode, the same treatment is given to the last record entered. The available track space (ATT/AVLTK) is set to a maximum.

   Whether or not a new track was needed, the beginning of record bit (ATT/SW1/2) is cleared and the TRB header is built. IDK mode bit (ATT/CMODE/14) is cleared and a check is made for required autosequencing. If the record is an insert, the autosequencing number is automatically set to 9999. If the record is not an insert, the autosequencing count (ATT/ASCNT) is converted to decimal (four bits per number) via ARITH (BD) and the ASCII autosequencing value is stored in the TRB. EMM (BOFFE) is called to reinitialize the ATT for the first field and the first field display is generated via DISPLY. EMM (STTYPE) is called to set in the ATT the correct data type from the format and EMM (AUTO) is called to perform any necessary automatic skip/dup requirements. Exit from BORENT is made via the common exit subroutine.

DOCUMENT CLASS **CYBERDATA IMS**

PROGRAM NAME **BORENT - Beginning of Record (Entry Mode)** **LEVEL 6**

INTERFACE

1. ENTRY METHOD

    Return jump

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address

3. EXIT METHOD

    Return to caller -- no error conditions

    JMP -- (ERROR) -- no disk space left

4. OUTPUT PARAMETERS OR CONDITIONS

    I = current ATT base address

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO (S12)
        ZERO (S22)
        ONEBIT (S23)
        ZROBIT (S33)
        TEN (S46)
        SST (S47)
        CLCDWA (S90)
        ENTSR (S91)
        EXTSR (S92)
        ERROR (S96)
        DISPLY (SA0)
        REWRIT (SA4)
        READP (SA5)
        READN (SA6)
        REDSEC (SA7)
        WRTSEC (SA8)
        CAM (SAE)
        STRTRY (SAF)
        WAIT (SB0)
        REDDWA (SA9)
        WRTDWA (SAA)

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___BORENT - Beginning of Record (Entry Mode)_____LEVEL 6

      b.   ATT

          FTOB (1)
          CMODE (2)
          FCNT (3)
          RCNT (4)
          ASCNT (5)
          ATRB (6)
          FRBA (7)
          PFBA (7)
          AVLTK (8)
          CURDA (9)
          NXTDA (11)
          CFFA (12)
          FMTNO (13)
          RFA (14)
          BMOD (15)
          BMDA (17)
          SW1 (19)
          SW2 (20)
          RCD (32)
          LUN (35)
          ATEMP (73)

      c.   CONSTANTS

          ERRF3 ($16)
          ERRF7 ($1F)

2.   MACROS USED

   None

3.   PROGRAM DEPENDENCIES

   PIO   - REDDWA, WRTDWA, REDSEL, WRTSEL
   ENEXSR   - ENTSR, EXTSR, WAIT
   GSE   - ERROR
   DISPLY
   CAM
   DAM   - GITRT, RELTRT
   ARITH   - BD
   EMM   - AUTO, BOFFE, STTYPE

4.   DATA AREAS USED

   ATT
   TRB
   DTL

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___BORENT - Beginning of Record (Entry Mode)_____ LEVEL 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $10D_{16} = 269_{10}$ words

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Reentrant.

AA5641

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME__ BORKEY __ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The program processes the BOR key-function, used in READ mode to backspace
    to the first field of the current record.

2. DETAILED DESCRIPTION

    The program first checks for legality of function (it is legal only in
    "pure" READ mode). In case the mode was not "pure" READ, exit is made via
    ERROR.

    Otherwise, the related pointers and counters are updated. The current
    record character count is also cleared. The relative FMT Field Address
    is also updated. A RTY is made to STTYPE (entry point in EMM) in order to
    set the data type.

    DISPLY is invoked and on return exit is made via CLRSR.

INTERFACE

1. ENTRY METHOD

    BORKEY is a disk resident program invoked by ENTPRG.

2. EXIT METHOD

    a. JMP - (ERROR) - in case the function is legal.

    b. JMP - (CLRSR) - normal exit.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. TRANSFER VECTOR

        ERROR ($96)
        CLRSR ($93)
        DISPLY ($A0)

    b. SST, ATT

        MODE (2)
        FCNT (3)
        RCNT (4)
        PFBA (7)
        RFA (14)
        SST ($47)
        CFFA (12)
        FLDN (14)
        ATEMP (73)

    c) LOCORE

        NZERO ($12)
        ZERO ($22)
        LPMASK (2)

2. EXTERNALS

    DBORKY
    STTYPE

3. MACROS USED

    None.

4. PROGRAM DEPENDENCIES

    ERROR
    CLRSR
    DISPLY
    STTYPE (entry point in EMM)

5. AREAS AND SYSTEM TABLES

    SST
    ATT
    FBA

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $23_{16} = 35_{10}$ words of program

2. RELOCATION TYPE

    Disk resident.

3. REENTRANCY

    Reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME__BSC__ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    BSC is a disk resident routine to process the BSC key. This function may be used in any mode, but not at the beginning of record.

    BSC has two entry points: BSCKEY, BSC2. It is loaded via ENTPRG when the operator hits the BSC key. Execution begins at the first entry point BSCKEY. Exit method is: JMP - (CLRSR) when normal termination occurred, and: JMP - (ERROR) with Q=4, when erroneous termination occurred.

2. DETAILED DESCRIPTION

    BSC first calls SETMOV to clear the program header I/O bit. If BSC is pressed at beginning of record, the routine exits to ILLEGAL FUNCTION KEY error. Further actions are taken according to the mode and the position in the record:

    a. In Verify, Pseudo Verify and Replace Field modes, and not at beginning of field, record and field counters are decremented by one. The current field is redisplayed via DISPLY - DBSCEV. In-sight-verify we redisplay via DISPLY - DBSCR to backspace the cursor and not clear the character. Exit is via CLRSR.

        If BSC is pressed at the beginning of field - jump to error (with Q=4).

    b. In READ mode:

        If BSC is pressed not at the beginning of field: record and field counters are decremented by 1; the cursor is placed on the previous character as the routine does a return jump to DISPLY - DBSCR and exits to CLRSR.

        If after backspacing the beginning of the record is reached, the routine sets IDK and SW1-B2 (first character of record next) bits.

        If BSC is pressed at the beginning of field:

        1) Number of words in previous format field is subtracted from current and relative format field address, so they point to the previous format field.

        2) Record-counter and field-number are decremented by one.

        3) Current field counter points to the last character of the previous field (which is now actually the current field).

- Data-type in ATT is updated - STTYPE.

- The cursor backspaces to the last character of the previous field as the routine does a return-jump to DISPLY - REGEN and exits - CLRSR.

c. In Entry and Insert modes:

If BSC is pressed not at the beginning of field: record and field counters are decremented by one. The routine redisplays the current field after backspacing and clearing last character, as it does a return jump to display - DBSCEV and exits to CLRSR.

If after backspacing the beginning of record is reached:

- Set IDK and SW1-B2 (first character record next) bits.

- If we are in Entry mode and not in the first record of batch we read the previous record, via RTJ - (READP) in order to update addresses and data in TRB. DUPKEY - bit SW2-B4 is not set.

  If not at BOR and in Entry mode and not in the first field in format, DUPKEY bit is set because the former data was destroyed.

(*) If BSC is pressed at the beginning of field: do 1), 2), 3)as in READ mode; we backspaced to the last character of the previous field. If we reached the end of the first field in the record and it is an auto. seq. field we pass to Beginning Of Record state and exit (see BOR).

Updating counters and checking error flags:

We exit BSC disk function (RTJ- (EXDFUN)) and jump to REPKEY (entry point 3) by: RTJ- (ENTPRG) with the proper parameters at Q- register in order to:

- Clear error bit (if it is set) of current field.

- If there is no other flagged field in reocrd, clear error bit in TRB header.

- Update the counters of field (if any).

BSC pass to REPKEY the difference between the record counter and the current field counter in bits 0-11 of ATEMP, so that only fields from the beginning of the record till current field will be checked for error flags.

Control returns to BSC at second entry point - BSC2.

If we are at the end of an automatic field (auto. dup. or auto. skip) and MLT. DUP/SKIP not pressed, we pass to the beginning of current field (updating record and field counters).

If we reached the beginning of record we set the flags and exit
(execute BOR from 2)). If not we loop and begin checks from (*).

If we are at the end of a nonautomatic field or MLT.DUP/SKIP was
pressed, we check whether we have to set DUPKEY bit. If we are in
Entry mode and not in the first field in format dupkey-bit SW2-B4 is set.
Then we check data-type-STTYPE, DISPLY - REGEN and exits.

BOR - Beginning Of Record:

1) If we are in Entry mode: decrement the auto. seq. number by one.

2) Set IDK and SW1-B2 (first character of record next) bits.

3) Set field number, record and field counters to zero.

4) Set relative format field address to 4 (points to the first field).

5) Update durrent format field address (in STT).

6) If not first record in batch and not inserted record we read
   previous record, via RTJ- (READP) to update address and data in
   the TRB. DUPKEY bit, SW2-B4 is not set.

7) Update data - type in ATT - STTYPE.

8) Redisplay - REGEN.

9) Exits: JMP - (CLRSR).

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

       LPMASK (2)
       NZERO ($12)
       ZERO ($22)
       ONEBIT ($23)
       ZROBIT ($33)
       SST ($47)
       CLRSR ($93)
       ERROR ($96)
       ENTPRG ($9E)
       EXDFUN ($9F)
       DISPLY ($A0)
       READP ($A5)

    b. SST

       CFFA (12)
       RFA (14)

c. ATT

    MODE (2)
    FCNT (3)
    RCNT (4)
    AUSQ (5)
    MISCH (5)
    ATRB (6)
    PFBA (7)
    SW1 (19)
    SW2 (20)
    STEMP (73)
    FLAG (ATEMP)

2. PROGRAM DEPENDENCIES

    SETMOV - Clear the program header I/O bit.
    ERROR   - Display error at terminal.
    DISPLY - Handle display output.
    CLRSR   - Free stack and exit to dispatcher.
    READP   - Read previous record.
    EXDFUN - Exit disk resident function.
    ENTPRG - Common program entry.
    STTYPE - Update data-type.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $E8_{16} = 232_{10}$

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

BSF is a disk resident routine to process the BSF key. BSF is illegal
in the following modes: Verify, Pseudo Verify, and in Replace field modes.

BSF has two entry points: BSFKEY, BSF2. It is loaded via ENTPRG when the
operator hits the BSF-key. Execution begins at the first entry point BSFKEY.

Exit method is: JMP - (CLRSR) when normal termination occurred, and:
JMP - (ERROR) with Q=4, when erroneous termination occurred.

2. DETAILED DESCRIPTION

BSF first calls SETMOV to clear the program header I/O bit. Then the
routine check the current mode. In Verify, Pseudo Verify and Replace
Field modes the routine exits to error. BSF is not allowed also at
the beginning of record in Entry and Insert modes.

a. in READ mode:

   ● If we are not in the first field of record the routine sets all
   pointers at the beginning of the previous field:

   1) Subtract field count and max. number of words in previous field
   from record count.

   2) Set field counter to zero.

   3) Number of words in previous format field is subtracted from
   current and relative format field address, so they point to
   the previous format field.

   4) Field number is decremented by one.

   5) Update data - type - STTYPE.

   6) DISPLY - DBSF.

   7) Exit - CLRSR.

   ● If we are in the first field of record, position to last field of
   the previous record:

   1) Subtract field count from record count. Record count points
   to the beginning of record.

   2) Set field count to zero.

   3) Rewrite record back on disk if corrections were made.

   4) Read previous record.

   5) Decrement record count in read by one.

   6) Get current format, via GFCORE.

7) Update current and relative format field addresses to point to the last format field of the previous record.

8) Record count get the difference between max. number of characters in the previous record and max. number of characters in the last field of the previous record, point to the beginning of the last field.

9) Update data-type, display and exit (see 5), 6) and 7)).

b. In Entry and Insert modes:

● If we are __not__ at first of format position to the beginning of first previous, nonautomatic field:

1) Current field counter is subtracted from record count.

2) Set field count to zero.

3) Max. number of words in previous field is subtracted from record count.

4) Field number is decremented by one.

5) Update current and relative format field addresses.

6) Updating counters and checking error flags:

We exit BSF disk function via EXDFUN and jump to REPKEY (entry point 4) via ENTPRG, in order to: clear error bit of current field.

- If there is no other flagged field in record, clear error bit in TRB header.

- Update counters of the field.

BSF, as BSC, pass to REPKEY the difference between the record and field counters in bits 0-11 of ATEMP, so that only fields from the beginning of the record till current field will be checked for error flags. Control returns to BSF at second entry point -BSF2.

7) If record count is equal to zero, pass to beginning of record state, display and exit (perform BOR).

8) If we are on nonautomatic field or MLT DUP/SKIP was pressed the routine set SW2-B4 (the DUPKEY-bit is set if we are in Entry mode and not in the first field of the format); check data-type via STTYPE; display -REGEN; and exits via RTJ- (CLRSR).

9) If we are on automatic field and MLT DUP/SKIP was not pressed we skip the field and position to the beginning of the previous field - jump to (3).

- If we are at first field of record we pass to beginning of record state:

  1) Current field count is subtracted from record count.

  2) Set field count to zero.

     BOR - Beginning Of Record.

  3) If in Entry mode, decrement auto. seq. number.

  4) Set IDK and SW1-B2 (first character of record next) bits.

  5) Set record count to zero.

  6) Update current and relative format field addresses.

  7) If we are not in the first record in batch and not in inserted record, read previous record, via READP (SW2-B4 not set).

  8) Update data-type - STTYPE, DISPLY - REGEN, and exits via CLRSR.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

       LPMASK (2)
       NZERO ($12)
       ZERO ($22)
       ONEBIT ($23)
       ZROBIT ($33)
       SST ($47)
       CLRSR ($93)
       ERROR ($96)
       ENTPRG ($9E)
       EXDFUN ($9F)
       DISPLY ($A0)
       READP ($A5)
       GFCORE ($9D)
       REWRIT ($A4)

    b. SST

       CFFA (12)
       RFA (14)

    c. ATT

        MODE (2)
        FCNT (3)
        RCNT (4)
        AUSQ (5)
        ATRB (6)
        PFBA (7)
        SW1 (19)
        SW2 (20)
        ATEMP (73)
        FLAG (ATEMP)
        RCORD (31)

2. PROGRAM DEPENDENCIES

    SETMOV - Clear the program header I/O bit.
    ERROR - Display error at terminal.
    DISPLY - Handle display output.
    CLRSR - Free stack and exit to dispatcher.
    READP - Read previous record.
    EXDFUN - Exit disk resident function.
    ENTPRG - Common program entry.
    STTYPE - Update data-type.
    GFCORE - Get current format.
    REWRIT - Rewrite record on disk if corrections were made.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $BB_{16} = 187_{10}$

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___BSFKEY___   Level 06___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

BSF is a disk-resident routine to process the BSF key. BSF is illegal in the following modes: Verify, Pseudo Verify, and in Replace Field.

BSF has two entry points: BSFKEY, BSF2. It is loaded via ENTPRG when the operator hits the BSF key. Execution begins at the first entry point BSFKEY.

Exit method is: JMP - (CLRSR) when normal termination occurred, and: JMP - (ERROR) with Q=4 when erroneous termination occurred.

2. DETAILED DESCRIPTION

BSF first calls SETMOV to clear the program header I/O bit. Then the routine check the current mode. In Verify, Pseudo Verify, and Replace Field modes the routine exits to error. BSF is not allowed also at the beginning of record in Entry and Insert modes.

a. In READ mode:

If we are <u>not</u> in the first field of record the routine sets all pointers to the beginning of the previous field:

1) Subtract field count and maximum number of words in previous field from record count.

2) Set field counter to zero.

3) Number of words in previous format field is subtracted from current and relative format field address, so they point to the previous format field.

4) Field number is decremented by one.

5) Update data - type - STTYPE.

6) DISPLY - DBSF.

7) Exit - CLRSR.

If we are in the first field of record, position to last field of the previous record:

1) Subtract field count from record count. Record count points to the beginning of record.

2) Set field count to zero.

3) Rewrite record back on disk if corrections were made.

4) Read previous record.

5) Decrement record count in read by one.

6) Get current format, via GFCORE.

7) Update current and relative format field addresses to point to the last format field of the previous record.

8) Record count get the difference between maximum number of characters in the previous record and maximum numbers of characters in the last field of the previous record, point to the beginning of the last field.

9) Updata data type, display and exit (see steps 5, 6, and 7).

b.   In ENTRY and INSERT modes:

If we are not at first of format position to the beginning of first previous, nonautomatic field:

1) Current field counter is subtracted from record count.

2) Set field count to zero.

3) Maximum number of words in previous field is subtracted from record count.

4) Field number is decremented by one.

5) Update current and relative format field addresses.

6) Updating counters and checking error flags:
   We exit BSF disk function via EXDFUN and jump to REPKEY (entry point 4) via ENTPRG, in order to:
   - Clear error bit of current field.
   - If there is no other flagged field in record, clear error bit in TRB header.
   - Update counters of the field.
   BSF, as BSC, pass to REPKEY the difference between the record and field counters in bits 0-11 of ATEMP, so that only fields from the beginning of the record till current field will be checked for error flags. Control returns to BSF at second entry point - BSF2.

7) If record count is equal to zero, pass to beginning of record state, display and exit (perform BOR).

8) If we are on nonautomatic field or MLT DUP/SKIP was pressed, the routine set SW2-B4 (the DUPKEY bit is set if we are in entry mode and not in the first field of the format); check data type via SSTYPE; display -REGEN; and exits via RTJ- (CLRSR).

9) If we are on automatic field and MLT DUP/SKIP was not pressed we skip the field and position to the beginning of the previous field - jump to (3).

If we are at first field of record we pass to beginning of record state:

1) Current field count is subtracted from record count.

2) Set field count to zero.
   BOR - Beginning Of Record.

3) If in Entry mode, decrement auto. seq. number.

4) Set IDK and SW1-B2 (first character of record next) bits.

5) Set record count to zero.

6) Update current and relative format field addresses.

7) If we are not in the first record in batch and not in inserted record, read previous record, via READP (SW2-B4 not set).

8) Update data type - STTYPE, DISPLY - REGEN, and exit via CLRSR.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)
      SST ($47)
      CLRSR ($93)
      ERROR ($96)
      ENTPRG ($9E)
      EXDFUN ($9F)
      DISPLY ($A0)
      READP ($A5)
      GFCORE ($9D)
      REWRIT ($A4)

   b. SST

      CFFA (12)
      RFA (14)

c. ATT

MODE (2)
FCNT (3)
RCNT (4)
AUSQ (5)
ATRB (6)
PFBA (7)
SW1 (19)
SW2 (20)
ATEMP (73)
FLAG (ATEMP)
RCRD (31)

PROGRAM DEPENDENCIES

SETMOV - Clear the program header I/O bit.
ERROR - Display error at terminal.
DISPLY - Handle disply output.
CLRSR - Free stack and exit to dispatcher.
READP - Read previous record.
EXDFUN - Exit disk resident function.
ENTPRG - Common program entry.
STTYPE - Update data type.
GFCORE - Get current format.
REWRIT - Rewrite record on disk if corrections were made.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $BB_{16} = 187_{10}$

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

CONTROL DATA CORPORATION • PSSD • SOFTWARE DOCUMENT

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ BSR - Backspace One Record _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   This routine processes the BSR key, when the terminal is in Read mode.

2. DETAILED DESCRIPTION

   BSR is a disk resident program. After being loaded, it starts with clearing the I/O bit through SETMOV.

   Mode is checked, and if it is not READ, ERROR is called with error code 4. If switch 1 bit 13 is set - for Rewrite record back or disk, REWRIT (in PIO) is called to perform writing. The previous record is read via READP (in PIO) and number of records in READ is decremented by 1. If the format of the new record is different from that in the ATT (format number is found in word 7 in TRB), the counters of number of characters in current field, and number of characters in current record, will be set to point to the beginning of the record. The new format is loaded via GFCORE. Current format field address in SST, and data type in ATT are updated.

   If the new record's format is the same as the last record, pointers will point to the same field in the new record as they pointed to in the last record when BSR was entered. The field being pointed to is displayed, and BSR exits via CLRSR.

INTERFACE

1. ENTRY METHOD

   BSR is loaded via ENTPRG when the BSR key is entered.

2. INPUT PARAMETERS OR CONDITIONS

   I = current ATT base address.

3. EXIT METHOD

   JMP - (ERROR) - with error code 4 when station is not in Read mode.
   JMP - (CLRSR) - when execution of function terminates normally.

4. OUTPUT PARAMETERS OR CONDITIONS

   I = current ATT base address.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME_____ BSR - Backspace One Record _____ Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)
      READP ($A5)
      DISPLY ($A0)
      CLRSR ($93)
      ERROR ($96)
      GFCORE ($9D)
      REWRIT ($A4)

   b. SST

      SST ($47)
      CFFA (12)

   c. ATT

      FMTNO (13)
      RFA (14)
      SW1 (19)
      RCDRD (31)
      MODE (2)
      FCNT (4)
      ATRB (6)
      PFBA (7)

   d. TRB

      TRBFMT (7)

2. MACROS USED

   None

3. PROGRAM DEPENDENCIES

   ENEXSR - CLRSR - Clears address stack in ATT and exits to dispatcher.
   GSE   - ERROR - Prints error messages.
   GFCORE   - Loads the format into core.
   PIO   - REWRIT - Writes a record back to disk.
             READP - Reads to the core the record previous to current record.
   DISPLY - DBSREL -
   SETMOV   - Clears I/O bit and enables moving the program.
   EMM   - STTYPE - Stores data type in ATT.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___BSR - Backspace One Record___   Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $(31)_{16} = (49)_{10}$

2. RELOCATION TYPE

   Run anywhere, movable.

3. REENTRANCY

   Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___CAM - Core Allocation Module___   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

There are three entry points in CAM: CAM honors requests for core space
by allocating a contiguous block of core in the Formats Records Programs
buffer area. (FRP). If a contiguous block of sufficient length is not
available, CAM attempts to create an adequate block by squeezing active
blocks down in the FRP buffer.

MMM moves an active block over a free area toward the beginning of the
FRP area. Only one block is moved on each call to MMM.

RPRD releases a designated block in the FRP.

Two additional subroutines in CAM are: RLSCOR, which builds a dummy block
header for a released area, and MMM1 which finds an active block and moves
it for MMM.

2. DETAILED DESCRIPTION

   a. CAM (For calling sequence see INTERFACE)

      1. Call ENTSR to save the return address in the exit stack.

      2. Check for more than MINCOR (=$800 words) available in allocatable
         area of FRP buffer. If so, go to step a5.

      3. If not, search the entire FRP buffer for sufficient continguous
         available space. This is done in the following 3 passes:

         I)   (CAM0I). The FRP is searched from the lower limit upwards for
              released buffers and for time-controlled programs with time
              over. The programs with time over are released till the core
              request is satisfied. A core request is satisfied if a buffer
              or the length of some subsequent buffers equal one or greater,
              at least with 3 locations from the amount of core required (the
              additional 3 locations are needed to build a dummy header for the
              remainder of the buffer).

         II) (NXTRAS). If a sufficiently large block was not found, the FRP
              is searched for the second time for the additional possibility;
              buffers will be moved, except for the buffers with I/O bits, if
              the sum of the areas surrounding these buffers will be sufficient
              for the core requested.

         III) If the above mentioned possibilities are not sufficient to get
              the amount of core required, the FRP is passed for the third
              time with the new additional possibility; time controlled
              programs which are not accessed but their time is still not
              over will be released if the length of these programs, together
              with the areas of the surrounding released buffers, are sufficient
              for the core requested.

PAGE NO. ___31___

4. (NXTOUT). If sufficient space is still not available, set
(A) = $FFFF, Q = required length, and call EXTSR to exit.

5. (OK). If the space is found, build the header for the block to
be assigned, adjust the FRPP pointer if necessary, and call RLSCOR
to release any excess space in the area found. Adjust the CFFA
pointer in SST, set (Q) = address of assigned block, (A) = 0 and
call EXTSR to exit.

6. (MOVE). If sufficient space was found in step 3II or 3III, call
MMM1 to move active blocks over the available areas (toward the
beginning of FRP).

b. MMM (For calling sequence see INTERFACE)

1. Set start address (SST/STEMP) = beginning address of FRP.
Set 'SM' release indicator (INDC) = 0.
Set FLAG=$FFFF.

2. Call MMM1 to move the next active block toward the beginning of
FRP.

3. If all active blocks have been handled, update the FRPP pointer to
the beginning of available space. Also reset SST/SLOCK/14 (storage
move flag) if no blocks have been skipped over.

4. Exit with (q) = 0.

c. RPRD (For calling sequence see INTERFACE)

1. Reset bit 15 (I/O control flag) and bit 14 (time control flag) in
word 1 of the block header.

2. Move the block disk address from word 3 of the block header to the
Program Directory (PRD) entry for the block.

3. Reset the program status bit for this block in the PRD control block.

4. Return to caller.

d. RLSCOR

1. Calling sequence is RTJ RLSCOR where:
Q = address of the block were from to release
SST+STEMP1 = length of area to be released.

2. Clear user count field in block header.

3. Put (block length-1) in length field.

4. If 2" block length, set SST/SLCOK/14 (storage move flag) and
exit to caller.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___CAM - Core Allocation Module_____Level 6

5. If (2'' + 2)　block length, put(2''-1) in length field, set remaining length=block length -2''.

6. Advance block pointer by length field + 1 and repeat from step 2.

7. If 2''　block length　(2'' + 2) put (2''-3) in length field, set remaining length=block length - (2''-3) and repeat from step 6.

e. MMM1

1. Calling sequence is RTJ MMM1 with starting address in SST/STEMP, a value in FLAG indicating the origin of the call (FLAG　0 if called from MMM, FLAG　0 if called from CAM), and required block length is LENGTH if the call is from CAM.

2. (MMM1). Save starting address in FREBAS.

3. (MMM3). Set length of free space = 0.
Move starting address from FREBAS to SST/STEMP.

4. (MMM6). If FRP block has been scanned up to FRPP, exit.

5. (GETNXT). Check the next block for activity. If user count = 0, time control is selected, and the block is preceded by　free space, go to step 13 to move the block. If the block is not preceded by a free space go to 9.

6. If user count = 0, and time control is not selected, update the length of free space by the length of this block.

7. If (FLAG)　0 (called from MMM), repeat from step 4.

8. If (FLAG)　0 (called from CAM), compare length of free space with required length. If free space length = required length, or if free space length　required length + 3, exit. Otherwise repeat from step 4.

9. If user count = 0, time control is selected and free space length = 0 repeat from step 3.

10. If user count ≠ 0, and free space length = 0, repeat from step 3.

11. If user count ≠ 0, I/O control is not selected, and free space length ≠ 0, go to step 13 to move this block.

12. If user count ≠ 0' I/O control is selected, and free space length ≠ 0, call RLSCOR to build a dummy header for the free space block, then repeat from step 3.

13. (MOVBLK). Move this block to the beginning of the current free space area. Update the address of free space in SST/STEMP by the length of this block.

14. If (FLAG) 0, call RLSCOR to build dummy header for the free space block, then exit.

15. If (FLAG) 0, repeat from step 4.

INTERFACE

1. CAM

CAM is a core resident routine whose entry point address is in SST location $AE.

Calling sequence is:

```
LDQ     Block length
RTJ-    (CAM)
   --Return--
I = ATT base address
```

Output parameters are:

```
(A) = 0 if space was allocated
    = $FFFF if not
(Q) = Beginning address of allocated space if space was allocated
    = Block length, if not.
```

2. MMM

MMM is core resident.

Calling sequence is:

```
RTJ MMM
--Return--
```

Contents of Q and A registers are destroyed.

3. RPRD

RPRD is core resident.

Calling sequence is:

```
LDQ     B         B = Address of disk resident program
RTJ     RPRD          to be released.
--Return--
```

Contents of Q register are preserved, A is destroyed.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)
      SST ($47)
      MONITR ($3C)
      ENTSR ($91)
      EXTSR ($92)

   b. SST

      SLOCK (1)
      PRDA (3)
      FRPLL (6)
      FRPUL (7)
      FRPP (8)
      CFFA (12)
      CLOCKS (17)
      STEMP (20)
      STEMP1 (21)
      STEMP2 (22)
      STEMP3 (23)

   c. ATT

      PFBA (7)
      RFA (14)

   d. CONSTANT

      MINCOR ($800)

3. PROGRAM DEPENDENCIES

   MONITR   Checkout monitor
   ENTSR   Enter subroutine
   EXTSR   Exit subroutine

4. DATA AREAS USED

   SST   System Status Table
   ATT   Active Terminal Table
   PRD   Program Directory
   FRP   Formats Records Programs buffer

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ CAM - Core Allocation Module_____Level 6___

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $13B_{16} = 315_{10}$ words.

2. RELOCATION TYPE

   CAM is a core resident program.
   It is in loader-relocatable form.

3. ENTRANCY CLASSIFICATION

   Not reentrant.

ATTACH ANALYTICAL SUPPLEMENTS

1. Formats/Records/Programs (FRP) buffer.

   Three words in SST define the FRP buffer. FRPLL contains the address of the
   first word of FRP, FRPUL contains the address of the first word adjacent to
   the end of FRP, and FRPP contains the address of the first word adjacent to
   the last active block in FRP. In other words, all the space from (FRPP) to
   (FRPUL) is free space.

2. FRP Block Header

   Each block of active and free space is defined by a 3-word header.

WORD 0	BACKWARDS POINTER			
1	1/0	T C			LENGTH
2				UC

AA5641

PRINTED IN THE USA

Where:

    I/O = I/O Control FLAG (1 = I/O in Progress)
    TC = Time Control FLAG (1 = Time Control)
    Length = Length of Block (not including backwards pointer)
            (11 bits)
    UC = User Count (6 bits)

A dummy header, as built by RLSCOR, has I/O = TC = 0, LENGTH =
length of block - 1, and UC = 0.

Backwards Pointer = 0
I/O = TC = 0
LENGTH = Length of block - 1
UC = 1

A header for the block just allocated by CAM has:

Active blocks may have additional header words following word 2 to
define disk address, time to release, etc.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___CANCEL___ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    To process the Cancel key after DLT and EOB.

2. DETAILED DESCRIPTION

    SETMOV is called to clear I/O bit and allow program move. CANCEL checks
    if bits 0 or 1 of Switch 1 are set, else ERROR is invoked with error code 4.
    If bit 1 in Switch 1 is set (for correct after delete to delete a record
    in Read and Verify), bits 11 and 1 in Switch 1 are cleared (11 - when set
    means that COR must be next). IDK bit in CMODE is reset. If mode is
    VERIFY, CANCEL will exit by RTJ (EXDFUN), and JMP+ VMM1; if Read mode,
    CANCEL exits to CLRSR. When bit 0 in Switch 1 is set (for EOB validation
    error, CANCEL, ERR O/R must be next), it is cleared. Switch 2 bit 12 is
    cleared (when it is set it indicates that ERR O/R may be entered). In this
    case CANCEL will exit to CLRSR.

INTERFACE

1. ENTRY MODE

    CANCEL is loaded via ENTPRG when the operator hits the CANCEL key.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

3. EXIT METHOD

    JMP - (ERROR) - with error code 4 when switch condition is illegal for cancel.
    JMP+VMM1   - (after EXDFUN), when COR/DLT in Verify mode.
    JMP- (CLRSR) - when COR/DLT in READ mode, or when EOB error flag is set.

4. OUTPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ONEBIT ($23)
        ZROBIT ($33)
        DISPLY ($A0)
        ERROR ($96)
        CLRSR ($93)
        EXDFUN ($9F)

    b. ATT

       SW1 (19)
       CMODE (2)
       SW2 (20)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

SETMOV - Clears I/O bit to allow program move.
DISPLY - REGEN - regenerates screen display.

4. DATA AREAS USED

ATT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$(24)_{16} = (36)_{10}$

2. RELOCATION TYPE

Run anywhere, movable.

3. ENTRANCY CLASSIFICATION

Reentrant.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME_____CBS - Change Batch Status_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    CBS is a Supervisor function used to change the present status of a batch
    to another status.

    Calling sequence is: CBS,JJJ,BBB,sss(,vvv) (CR)
    where: JJJ - Job name
            BBB - Batch number
            sss - The requested status
            vvv - when VER status is requested vvv is the verification
                    options requested

2. DETAILED DESCRIPTION

    CBS first enters SETMOV to enable program moving, and then collects the
    entered parameters via GETALP and GETBIN in SCANV. If a comma does not
    follow, the CBS command goes to REJSUP to indicate "invalid command format"
    (ERR9). If there are more than 6 characters in the job name or they are
    not legal characters or the job name is not followed by a comma, exit is
    to REJSUP to indicate "illegal job name" (JERR). For a batch number that
    exceeds 999 or is not followed by a comma, goes to REJSUP indicating
    "invalid batch number" (BERR). The requested "change" is checked to be
    OPN, ORV or VER; if another mnemonic was entered goes to REJSUP to indicate
    "invalid parameter" (OERR). When VER is requested as the change, the new
    verification options are expected - N for no verification options or any
    combination of FEU options. If any of these are not given goes to REJSUP
    to indicate invalid verification option (ERR17). Then prepares parameters
    and enters GETLJD to ensure that the job specified is a legal job and to
    extract the job index.

    Next a check is made to see if modifications are being made to the ABT. If
    this is the case (SST/SLOCK/13=1), exits to WAIT, and tries again. When no
    changes are being made, sets SST/SLOCK/13, and inserts job parameters in a
    request to SPILL to locate the specified batch in the ABT. If the batch does
    not exist in the ABT, goes to BERR; else saves the parameters returned by
    SPILL and performs the following: if the requested change is ORV, and the
    present mode is waiting to be verified, changes the mode to completed, and
    goes to update the ABT (UDABT); else goes to BERR. If the requested change
    is VER and the batch mode is not entry or entry forzen or verify changes it
    to verify frozen, inserts the new verification options and goes to UDABT;
    else goes to BERR. If the requested change is OPN and the batch mode is
    waiting to be verified or completed or written (if not one of these goes to
    BERR), issues to the disk associated with the batch a READ request. If the
    request is rejected goes to REJSUP to indicate "disk X is deselected". If the
    disk is selected, request 32 core words via CAM and reads via REDDWA 32 words
    of the ATT written on the disk. If no core is available, SST/SLCOK/13 is
    cleared, and exit to REJSUP with a suitable message. Words 5, 6, 7, 13, 14,
    27, 28 of the ATT are cleared, current terminal mode is set to IDK-IDLE, ATT/
    SW1/2,3,10 and ATT/SW2/3 are set to 1, ATT/basic mode is set to ENTRY and

writes back the 29 words via WRTDWA. Then transfers the batch mode in the
ABT to entry frozen and releases the requested core. 4 words of the last
record header of the batch are read via REDDWA. The disk address of the
next record in the header is updated according to the saved ATT and the
4 words are written back via WRTDWA (UDABT). Then the ABT is updated with
the new information via SPILL and exit is to SV, after clearing SST/SLOCK/13.

INTERFACE

1. ENTRY METHOD

   CBS is a supervisor function loaded by ENTPRG.
   Only entry point is CBS.

2. INPUT PARAMETERS OR CONDITIONS

   I = current ATT address
   Operator enters
           CBS,JJJ,BBB,ccc(,vvv) (CR)

3. EXIT METHOD

   1. Jump to REJSUP with appropriate error code

   2. Jump to SV

4. EXIT PARAMETERS OR CONDITIONS

   I = Current ATT address

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK ($2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)
      SST ($47)

   b. SST

      STEMP (20)
      SLOCK (1)

c. TRANSFER VECTOR

    CLCDWA ($90)
    ENTSR ($91)
    EXTSR ($92)
    GETBIN ($99)
    GETALP ($9B)
    GETSOM ($9C)
    SPILL ($A1)
    REDDWA ($A9)
    WRTDWA ($AA)
    CAM (SAE)
    SV (SB1)
    REJSUP ($B2)

d. EST

    DSKST (29)

e. ATT

    FTOB (1)
    JIDX (3)
    SC (36+10)
    TJOB (SC)
    TBATCH (SC+3)
    OPT (SC+4)
    OPTV (SC+5)
    WORD (SC+6)
    WORD2 (SC+7)
    WORD3 (SC+8)
    WORD4 (SC+9)

f. CONSTANTS

    E ($45)
    F ($46)
    N ($4E)
    U ($55)

3. PROGRAM DEPENDENCIES

    a. EXTERNALS

    SETMOV -
    GETLJD - Locates a job in the LJD.
    BD   - Converts hexadecimal number to BCD.

b. TRANSFER VECTOR

    ENEXSR - ENTSR   - Common subroutine entry.
                EXTSR   - Common subroutine exit.
    PIO   - CLCDWA   - Calculates disk word addressing.
                REDDWA - Read from disk according word addressing.
                WRTDWA - Write on disk according to word addressing.
    CAM   - Allocate core buffer.
    SCANV   - GETBIN - Get binary number from input string.
                GETALF - Get next alph character from input string.
                GETCOM - Get comma or end of record from input string.
    SUPER   - SV   - Conclude supervisor request.
                REJSUP - Generate error messages to supervisor.
    SPILL   - Handle Active Batch Table.

4. DATA AREAS

    SST
    EST
    ATT
    ABT
    Buffer requested via CAM to hold the disk resident ATT.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $18B_{16} = 395_{10}$ for the program.

    32_{10} words buffer as mentioned above.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Partially reentrant. If ABT modifications are being carried out, the
    main part of the program is locked.

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   CDS is a supervisor function to change the status of a disk drive from up-to down status or vice versa. Command format is CDS, logical unit, status.

2. DETAILED DESCRIPTION

   a. (CDS). Call SETMOV to allow the program to move in FRP buffer. If character after CDS in the command is not a comma exit to REJSUP with A = 9 (will display INVALID COMMAND FORMAT and allow new supervisor entry). GETBIN is called to get the logical unit which is stored in ATT/SAVLU. If logical unit is not followed by comma or not between 1 and 4 command format is improper and exit to REJSUP with A = 27 is made (will display INVALID LOGICAL UNIT and allow another supervisor entry). GETALP is called to get the status character (S or D) of the command and it will be stored in ATT/ATEMP. If the character was NONASCII, comma or EOR or not S or D exit is made to REJSUP with A = 20 (INVALID STATUS OPTION).

   b. (IFEOR). In GETCOM next character is retrieved from command buffer. If it is not an EOR character, exit is made to REJSUP with A = 20 (INVALID STATUS OPTION).

   c. (CHCKST). If status character stored in ATT/ATEMP was an S go to step f to mark unit as up.
      *Down status.

   d. Get EST/DSKST (disk status word). (C1) If it is legal to declare disk as down go to e. If all other disks are declared down, exit by calling subroutine MOVPRT. This will display the message "ALL OTHER DISKS DESELECTED INVALID CMD" on the supervisor list unit. Upon return exit with A = FFFF to REJSUP (will not display a message) and allow a new supervisor entry.

   e. (TSTATT). Get EST/DSKST. Mask out up/down bits and set the status bit for this disk to down (=1). Store updated entry back into EST/DSKST. Go to step k.

   f. (UP). Check EST/LUND1 + n (where n is disk number 0-3) to get logical unit number of disk to mark as up. If zero or 7FFF go to step g. Store logical unit of disk into RQLU of READ-request block. If PHYSTAB/EREQST bits 11-13 equals 2 (mass storage class code) go to step g1.

   g. (LUER). Set A = 27 and exit to REJSUP (will display "INVALID LOGICAL UNIT" and allow another supervisor entry).

      1) Get TRT length for the disk to be marked up from the corresponding byte of one of the first two words of EST. If the TRT table is more than one word long go to step h (there is space on disk that is to be marked as up). Otherwise A is set to 30 and REJSUP is called (will display "ERROR IN SYSTEM TABLES***" and allow another supervisor command to be entered).

h. (READ1). Find out absolute address of label READ1+1 in FRP buffer. Calculate hence the completion address and store in CMPL of READ-request block. Set buffer address to read one word into as ATT/BUF and store into BUFADD, call subroutine IOONF to set function lock and I/O-move bit. Issue a read request to read one word from sector one on the disk in order to verify its proper functioning. Go away to dispatcher.

i. (ERRBT). Upon completion of the diskread, restore the I register from SST/ATTB (terminal 0 is always at the ATT base address). Call subroutine IOONF to toggle back function lock on the ATT and reset the I/O-move bit for the module. If the diskread completed with no errors (Q positive) go to step j. If errors get the number of the existing disk, convert to ASCII code, plug into error-message and call subroutine MOVPRT to display "READ ERROR IN DISK X". Set A = 23 and go to REJSUP to display "I/O ERROR" and wait for another supervisor command.

j. (NOERR). Change EST/DSKST word to reflect that status for this disk is now up.

k. (WRTEST). Get address of EST/DSKST and plug into diskword write at label CORAD. Get sector address of the start of core-image via MSOS Extended communication region table convert to a word address through call to CLCDWA. Write the changed EST/DSQST word back to core image on system disk. Exit to SV to allow another supervisor entry.

SUBROUTINES

l. (MOVPRT). Call ENTSR to put return address into entry/exit stack. Move number of words of message to output into request blocks. Get starting address of message and plug into FROM. Store the start of supervisor interrogate buffer into TO. Call MOVREC to move message into IRB. Call SETMOV to set function index to allow a relative return during the I/O. Call DUPRW to display the error message in the IRB. Call RTNSR to get return address and exit out of subroutine.

m. (IOONF). Toggle ATT/0/14 (function lock) and bit 15 of first reader word of the program. Return to caller.

INTERFACE

1. ENTRY METHOD

CDS is a disk resident supervisor function which is called via ENTPRG when requested by DSKSUP.

2. INPUT PARAMETERS CONDITIONS

I = current ATT address.
Command in interrogate buffer.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____CDS - Change Disk Status_____Level 6

3. EXIT METHOD

Jump to REJSUP in order to display error message.
Jump to SV in order to get another command.

4. OUTPUT PARAMETERS OR CONDITIONS

If disk exists the disk status word in EST and the corresponding word in
the core image on disk is updated.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

       LPMASK ($2)
       NZERO ($12)   (not used)
       ZERO ($22)
       ONEBIT ($23)
       ZROBIT ($33)
       SST ($47)

   b. SST

       ATTB (10)
       ATTLNG (19)   (not used)
       STEMP (20)
       CLCDWA ($90)
       ENTSR ($91)
       EXTSR ($92)   (not used)
       SUPRW ($97)
       RTNSR ($98)
       GETBIN ($99)
       GETALP ($9B)
       GETCOM ($9C)
       WRTDWA ($AA)
       SV ($B1)
       REJSUP ($B2)
       DISPCH ($EA)

   c. EST

       LUND1 (25)
       DSKST (29)

   d. ATT

       BMODE (15)   (not used)
       AINT (27)
       DLU (35)   (not used)
       ATEMP (73)

e. SCA

SAVLU (50)
ATTLIM (51)   (not used)
BUF (52)

f. Constants

EREQST (8)   Request status word in PHYSTAB
D ($44)   Character D
S ($53)   Character S
FW (3)   Code for formatted write
EXTBL ($E9)   Pointer to extended system table
CORIMG (4)   Sector address of core image

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

SETMOV (ENEXSR)   Allows programs to move in FRP
MOVREC (PIO)   Moves messages to IRB
LOGIA (SYSDAT)   Address table of PHYSTAB's
ENTSR (ENEXSR)   Common routine to save return address
RTNSR (ENEXSR)   Recover last entry in address stack
GETBIN (SCANV)   Converts a numeric value in IRB to binary
GETALP (SCANV)   Checks character in IRB to be alpha
GETCOM (SCANV)   Checks character in IRB to be comma or EOR
CLCDWA (PIO)   Calculate a disk word address
WRTDWA (PIO)   Write in word address mode to disk
SV (SUPER)   Supervisor command processor
REJSUP (SUPER)   Supervisor reject procedure
SUPRW (PIO)   Read/write processor for supervisor console

4. DATA AREAS USED

SST   System Status Table.
ATT   Active Terminal Table.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$D1_{16} = 209_{10}$ words.

2. RELOCATION TYPE

Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CDVTST__

PROGRAM NAME___Check Digit Validation__

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    CDVTST performs End of Field check digit validation where the right-most digit in the field must have a specific relationship to the result of a combination of the remaining digits with prescribed weight values. The test is valid only for numeric fields of length 2-16, including a check digit as the right-most digit.

2. DETAILED DESCRIPTION

    a. (CDVNUM). Save the Check Digit Validation (CDV) code in INDEX.

    ● Compute the address of the designated CDV Weight Table (CWT) entry and save it in ENTCWT.

    ● If the field length, N, is 1 digit or less, go to step h.

    ● Compute address of last word of data field.

    b. (LOUP1). Move digits from data field to BUF + 16 through BUF + N + 15, reversing the order of the digits, that is, putting right-most digit in BUF + 16, next digit to the left in BUF + 17, and so forth. The check digit is in BUF + 16.

    c. (SOF). Move weights from CWT entry to BUF through BUF + N - 1, one digit per word. The modulus is in BUF.

    d. (DOOP). Check CDV code. If Test number 1 requested, go to step i.

    ● If Test number 3 requested, go to step j.

    e. (OP1). Multiply each data digit by the corresponding weight value, accumulating the products in FDCNT.

    ● If the modulus in BUF is zero, go to step h.

    f. (CHECK). Divide the weighted sum by the modulus value.

    ● Add the check digit to the remainder; if the result is zero, go to step g.

    ● If not, subtract the result from the modulus.

    g. (EXIT). If the result is nonzero, go to step h.

    If the result is zero, exit to caller.

    h. (ERR).Exit to ERROR with (Q) = $A (Check Digit Error).

    i. (SEV).Divide base number by the modulus.

    ● Exclusive OR the result with the check digit and go to step g.

,239

j. (DIXI). Multiply each data digit by the corresponding weight value, and divide the result by 10; hold the result in FDCNT. Go to step f.

INTERFACE

1. ENTRY METHOD

   EXT   CDVNUM
   RTJ   CDVNUM

2. INPUT PARAMETERS OR CONDITIONS

   (I) = Base address of Active Terminal Table (ATT)

   (A) = Check digit validation code. Currently defined validation procedures are:

   (A) = 0: Divide the value in the data field by 7. The remainder must equal the check digit.

   (A) = 1: Sum the digits in the data field. This result subtracted from the next higher multiple of 9 must equal the check digit.

   (A) = 2: Starting from the right, multiply each odd digit in the data field by 2; sum the resulting digits along with the even digits in the data field. This result subtracted from the next higher multiple of 10 must equal the check digit.

   (A) = 3: Multiply each digit in the data field by the corresponding weight values; divide the sum of the weighted digits by 11. The remainder subtracted from 11 must equal the check digit.

3. EXIT METHOD

   Normal exit: Return to location immediately following RTJ CDVNUM

   Error exit: Exit to ERROR with (Q) = $A

4. OUTPUT PARAMETERS OR CONDITIONS

   (I) = Base address of ATT

   (Q) = 0

   (A) = Undefined

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK ($2)
      NZERO ($12)

240

ONEBIT ($23)
ZROBIT ($33)
TEN ($46)
SST ($47)
ERROR ($96)

b. SST

CATT (11)
STEMP1 (21)

c. EST

ACWT (2)

d. ATT

FCNT (3)
RCNT (4)
ATRB (6)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

GSE (ERROR) Display error message

4. DATA AREAS USED

SST   System Status Table
ATT   Active Terminal Table
CWT   CDV Weight Table
TRB   Terminal Record Buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$AF_{16} = 175_{10}$ words

2. RELOCATION TYPE

Core resident, loader relocatable

3. ENTRANCY CLASSIFICATION

Not reentrant

DOCUMENT CLASS ___CENDS_____

PROGRAM NAME_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   CENDS is core resident with four entry points:

   a. CEOF   Display End-Of-Field
   b. CBOF   Display Beginning-Of-Field
   c. CEOREV   Display End-Of-Record ENTRY/VERIFY
   d. TUTMOD   Tutorial Display (if required)

2. DETAILED DESCRIPTION

   a. (CEOF). Perform ENDSR. Calculate the address of the next screen
      line and store it as the new current screen position in TOPT/CURSRD.
      Return to caller via EXDSR.

   b. (CBOF). Perform ENDSR. Reset current screen address in TOPT/CURSRD
      to beginning of line. Check ATT/SW1/3 to see if Doc., Form. Adv.,
      or F. SELECT must be keyed in. If so, go to CBFEX. Get screen address
      of End of Data Area (UPDAT) and subtract current screen position
      (CURSD) and subtract the number of characters in Field Name/Number.
      Increase this number by -2 (place for '**' error flag) and save as
      the number of available positions. Obtain the field length of the
      next field from the format (SST + FFA) and compare the number of
      positions needed against the number of positions available. If sufficient
      go to CBOF1, otherwise clear screen (with CLL), display the header
      (with CHD) and go to CBOF2.

      (CBOF1). RTJ to CFNDTY to display the field number.

      (CBOF2). RTJ to TUTMOD to do the tutorial mode if required. Then,
      extract the field number from ATT/FLDN and pass it as a parameter to
      CFDNO which displays the field number at the beginning of the line.

      (CBFEX). Increase TOPT/CURSRD by 2 to allow for error flag positions
      and return to caller via EXDSR.

   c. (CEOREV) Perform ENDSR. Then set ATT/RQ to $FF. RTJ to CPAGE
      to clear the screen and regenerate the header. RTJ to TUTMOD to
      perform tutorial display, if required. Then exit to caller via EXDSR.

   d. (TUTMOD). Perform ENDSR. Check ATT/SW1/4 to see if tutorial mode
      needed and if not, return to caller via EXDSR. Then check ATT/FMTNO
      and if zero, exit via EXDSR. Otherwise, save exit (SST/STEMP) in
      ATT/RQ + C. Begin to build tutorial message of the form "KEY FIELD
      FFFFFF WITH 99 FIX ALP CHARACTERS". First, set field name to all
      blanks. RTJ to GETBUF with A = 44 to get a message buffer. Restore
      exit from ATT/RQ + C back to SST/STEMP. Save the current SST/FFA

135

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME____CFS - Change Format Status _____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    CFS is a supervisor function to change the status of a format between active and inactive status.

    If parameters are illegal the format is not updated.

2. DETAILED DESCRIPTION

    a. (CFS). Call SETMOV to enable CFS to be moved, if necessary.

        1) If the character following CFS in the IRB buffer is not a comma go to step b.

        2) If the format number obtained in IRB is not numeric go to step b.

        3) If the format number is zero go to step c.

        4) Save the format number in ATT/47.

        5) If this format number is greater than the maximum allowed in EST/24 go to step c.

        6) If next character in IRB is an ASCII character go to step d.

    b. (ERR 9). Set (A) = 9 and exit to REJSUP to issue the message INVALID COMMAND FORMAT and wait for another supervisor command.

    c. (ERR 7). Set (A) = 7 and exit to REJSUP and issue the message INVALID FORMAT NUMBER and wait for another supervisor command.

    d. (TSTAI). If the next to last character in IRB (status) is A go to step e.

        1) If the status character is I go to step f.

        2) Set (A) = 20 and exit to REJSUP and issue the message INVALID STATUS OPTION and wait for another supervisor command.

    e. (SETA). Set ATT/Status = 0 to reflect active status go to step f3.

    f. (SETI). Set ATT/Status = $8000 to reflect inactive status.

        1) If last character in IRB is not EOR go to step b.

        2) Set (A) = EST/39 (LFD base address = sector)
              (Q) = (ATT/47)/2 (format number = word displacement).

        3) Call subroutine READ1 to read the LFD entry for this format number.

AA5641

4) If the format address is nonzero go to step f6 (the format exists).

5) Set (A) = 28 and exit to REJSUP and issue the message UNDEFINED FORMAT NUMBER and wait for another supervisor command.

6) Call subroutine WRITE1 to update the format status in LFD entry.

7) Set (A) = ATT/49 (format sector is base address).
    (Q) = 1 (word displacement in sector is one).

8) Call subroutine READ1 to read last two words of format header.

9) Call subroutine WRITE1 to update format status in the format entry.

10) Exit to SV.

SUBROUTINE READ1

g. (READ1). Read two words from disk into ATT/BUF using the sector address from A and word-displacement in Q.

1) Save return address in exit stack.

2) Call CLCDWA to convert to a proper word address.

3) Plug word address into request block (DWA1,DWA1+1)/

4) Save this word address also in ATT/ADWA2,ADWA2+1.

5) Calculate absolutized buffer address of where to read into (ATT/BUF) and plug into request block (BUFAD1).

6) Call SETMOV to allow move during I/O.

7) Call REDDWA to read from disk (2 words).

8) Call EXTSR to exit to the last address in exit stack (return address of subroutine).

SUBROUTINE WRITE1

h. (WRITE1). Write two words to disk from ATT/BUF using a disk word address in ATT/ADWA2.

1) Save return address in exit stack.

2) Update the format status bit of ATT/BUF + 1/15 with ATT/STATUS/15.

3) Plug the word address into request parameters from ATT/ADWA2 to DWA2 and ATT/ADWA2 + 1 to DWA2 + 1.

4) Plug address of ATT/BUF into BUFAD2.

5) Call SETMOV to allow moving during I/O.

6) Call WRTDWA to write to disk (2 words).

7) Call EXTSR to exit to the last address in exit stack (return address of subroutine).

    b. SST

        STEMP (20)

    c. EST

        MAXFMT (24)
        LFDDA (39)

    d. ATT

        FORMAT (47)
        STATUS (48)
        BUF (49)
        ADWA2 (51)

    e. ALPHA CHARACTERS

        SA ($41)   letter A
        SI ($49)   letter I

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

PIO	CLCDWA	Calculate disk word address
	REDDWA	Read disk word address
	WRTDWA	Write disk word address
SCANV	GETBIN	Get characters up to comma and convert to binary number
	GETALP	Get next alpha character from input string
	GETCOM	Get comma, EOR or nonspace char. from input string
SUPER	REJSUP	Supervisor reject procedure
	SV	Supervisor command processor
ENEXSR	SETMOV	Enable disk resident program to move
	ENTSR	Save return address in exit stack
	EXTSR	Exit to last address in exit stack

4. DATA AREAS USED

SST	System status table
ATT	Active terminal table
LFD	

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __CFS - Change Format Status__ _____ Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $5E_{16} = 94_{10}$ words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA IMS_____

PROGRAM NAME___CIO - Change I/O Units_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   This function is a supervisor function to change input and/or output devices of supervisor/terminal.

   Parameters are - CIO,Ixx,Lyy,Pzz

           xx - Input Logical Unit preceded by I
           yy - Output Logical Unit preceded by L
           zz - Mag Tape Logical Unit preceded by P

   Three or less parameters may follow the CIO command but at least one parameter must be entered.

   When comma or ZERO follows the preceding letters, it means return to basic logical unit of the terminal.

2. DETAILED DESCRIPTION

   CIO receives control from DSKSUP, after storing the CIO command in the Interrogate buffer. It scans the command and updates the Input/List/ Tape/Unit in the ATT (words 36-38) if the parameters are O.K.

   a. The function checks for I, L or P. If not one of these letters, exit via REJSUP (INVALID COMMAND FORMAT).

      If the function has been asked for by a terminal, then only P is allowed as a parameter. Else exit via REJSUP (COMMAND INVALID ON THIS STATION).

   b. The function checks the logical unit number after the letter. If none of ZERO, return to basic logical unit is performed by replacing INPUT/ LIST/TAPE/UNIT (words 36-38 in ATT) with the BASIC INPUT/LIST UNIT (words 39, 40 in ATT) or logical unit of mag tape.

      Before doing it checks if it is not the same device which has been asked. Then checks for end of command and exits via SV, else repeats from step a.

   c. If the logical unit number is nonzero, the function checks the number against max. logic unit (LØGIA); if greater, exits via REJSUP (INVALID LOGICAL UNIT); else, checks equipment class where:

              Tape unit is legal for I/L/P,
              Card device is legal only for I
              Printer device is legal only for L
              Paper tape device and Teletype are valid for I and L

      Else exits via REJSUP.

AA5641

d. CIO does not allow more than one I/O unit at a time for mag tape; if so, exits via REJSUP.

If the preceding letter of the parameter is P, checks if device was already assigned to: TTY input unit, or TTY list unit, or new assignment to: TTY input unit or TTY list unit and if so, exits via REJSUP.

Else, checks if there is another supervisor station which uses the same logical unit; if so - exits via REJSUP (DEVICE BUSY).

Else, checks if there is one more parameter. If yes, keeps logical unit number of current parameter and passes to next parameter (Step a.); if there are more than three parameters, exits via REJSUP.

If last parameter, updates the Input/List Tape/Unit, then exits via SV.

INTERFACE

1. ENTRY METHOD

   CIO is a disk resident program which is invoked from DSKSUP by ENTPRG.

2. INPUT PARAMETERS

   (I) = Base address of Active Terminal Table (ATT).

3. EXIT METHOD

   Exit is to REJSUP if an error is detected, to SV otherwise.

4. OUTPUT PARAMETERS

   (I) = Base address of ATT

   Input Unit - ATT word 36
   Output Unit - ATT word 37
   Tape Unit   - ATT word 38

ASSEMBLY CHARACTERISTICS

1. EQUATES

   SCAI(36)
   SCAO(37)
   SCAT(38)
   BLU(39)
   BUF(50), TEMO1(57), SAVTYP(53), SAVELU(54), INDEX(55), LI($49),
   LL($4C), LP($50), GETALP($9B), GETBIN($99), REJSUP($B2), SV($B1),
   LPMASK($2), ZERO($22), ØNEBIT($23), ERQST(8), ATTLIM(56)

2. PROGRAM DEPENDENCIES

SUPER $\begin{cases} \text{SV} \\ \text{REJSUP} \end{cases}$

SCANV $\begin{cases} \text{GETALP} \\ \text{GETBIN} \end{cases}$

3. EXTERNALS

LOGIA

4. DATA AREAS USED

SST	-	SYSTEM STATUS TABLE
SCA	-	SUPERVISOR COMMUNICATION AREA
LOGIA	-	LOGICAL UNIT TABLE
PHYSTB	-	PHYSICAL DEVICE TABLE

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE

$99_{16} = 153_{10}$ for program.

2. RELOCATION TYPE

Run anywhere.

3. REENTRANCY

Reentrant.

REFERENCES

1. CYBERDATA System Reference

2. CYBERDATA Software User's Guide

3. MSOS Version 4 Reference Manual

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   CLEARS has two entry points:

   CCLRFD clears the interrogate line, or clears a field or a line in the Data area of the screen.

   CCLRAL clears the entire screen.

2. DETAILED DESCRIPTION

   a. CCLRFD

      1) (CCLRFD). Call ENDSR to save return address in Exits stack.

         ● If in data area of screen, go to step a3.

         ● Set cursor position to beginning of Interrogate line.

      2) (CLR01). Call CDATA to clear line and display cursor.

         ● Exit to caller via EXDSR.

      3) (CLR1). If ATT/RQ 0 (Backspace to beginning of line), set cursor position to beginning of current line, and go to step a2.

      4) (CLR10). Calculate address of 1st line of the field and save it.

         ● Calculate address of last line of the field and save it.

      5) (CLR12). Clear field line via CDATA.

         ● If the last line of the field was cleared go to a6. Else, advance screen address in parameter list to next line and go to a5.

      6) (CLR15). Call CFDNO to display current field number.

         ● Advance cursor position by 2 to bypass error flag positions and exit to caller via EXDSR.

b. CCLRAL

1) (CCLRAL). Call ENDSR to save return address in Exits stack.

    ● Reset screen addresses to first position of data area and
      first position of interrogate line.

    ● If there is no display string to be output, go to step b3.

    ● Return all but the last block of the display string to the
      string of free blocks.

2) (KCLR12). Set all words of remaining block to zero.

    ● Set block count to 1.

3) (KCLR2). Call CDATA to clear the screen.

    ● Exit to caller via EXDSR.

INTERFACE

1. ENTRY METHOD

    EXT   A   Where A = CCLRFD or CCLRAL
    RTJ   A
    --Return--

2. INPUT PARAMETERS OR CONDITIONS

    (I) = Base address of Active Terminal Table (ATT)

    CCLRAL requires no other input parameters.

    CCLRFD requires an additional parameter in ATT/RQ.

    If (ATT/RQ) $\geq$ 0, clears current field.

    If (ATT/RQ) $<$ 0, clear current line.

3. EXIT METHOD

    Return to location immediately following RTJ instruction.

4. OUTPUT PARAMETERS OR CONDITIONS

    (I) = Base address of ATT

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ CLEARS - Clear a Field on Display _____ Level 06

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        SST ($47)
        CDATA ($8D)
        ENDSR ($8E)
        EXDSR ($8F)

    b. SST

        CFFA (12)

    c. ATT

        FLDNO (14)
        TOPT (33)
        RQ (65)

    d. TOPT

        COL (2)
        CURSRD (3)
        CURSRI (4)
        PARAM (8)
        TOPSTR (13)
        BLOKS (14)
        FLDST (15)

    e. CONSTANTS

        BLOKF (3)
        BLOKL (8)

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___CLEARS - Clears a Field on Display_____Level 06

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   CMISCI ENDSR   Save return address in Exits stack
           EXDSR   Return to last address in Exits stack

   VL2CRT (CDATA)   Display message on CRT

   HDRNOS (CFDNO)   Display field number

4. DATA AREAS USED

   SST   System Status Table
   ATT   Active Terminal Table
   TOPT   Terminal Output Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $75_{16} = 117_{10}$ words.

2. RELOCATION TYPE

   Core resident, loader relocatable.

3. ENTRANCY CLASSIFICATION

   Not reentrant.

DOCUMENT CLASS ___CLR_____

PROGRAM NAME_____Extended Power Fail Recovery (1)_____

PROGRAM FUNCTION

Entry points: CLR, ENT2

1. GENERAL DESCRIPTION

CLR program releases tracks which were used by certain system routines which were working when power fail occurred.

The CLR program checks if active batches can be restored to the stations they were connected to before the power fail occurred. If a batch cannot be reactivated it is put into the frozen state. If a batch is in ENTRY mode and has less than 16 records it is deleted. If reactivation of certain batches is possible, the CLR2 program is brought into memory and is executed. Meanwhile CLR is released. CLR2 restores the batch to its station and returns to CLR.

Return is done by loading CLR program into memory and executing it from the second entry point ENT2, where scanning of the ABT's is done.

2. DETAILED DESCRIPTION

a. Asks for interrogate buffer, 35 words long.

a1. The Bad Tracks Directory in DAM is scanned. If nonempty entry is found corresponding to track address of disk n (1-4) the message: BAD TRACK IN DISK n. CLEAR DIRECTORY (/N) is printed. If the supervisor wishes to clear the entries corresponding to disk n, in case the disk pack was exchanged she answers   . The entries whose 2 LSB equals n-1 are cleared. If an attempt was made before release a bad track, bit 3 in the entry is set. The track is then released via RELTRT (subroutine of DAM).

b. Releases tracks listed in the Tape Recovery Area unused words, and clears the list.

The tracks are written in this list by the system routine PST (Print Statistics). The routines use them for processing.

c. The Active Batch Tables (ABT's) are read into memory one after the other. Each ABT is checked.

(i) Unused entries are skipped.

(ii) Batches in maintenance mode,i.e. were being processed by any of the dump routines when power fail occurred, are assigned their original modes

(iii) Batches in ENTRY or VERIFY mode get special treatment.

    1) The logical unit, the address of first track and the job index are transferred from the ABT entry to the ATT of the executing station.

        1a) If the logical unit "deselected" bit is set the batch cannot be reactivated. A flag is set indicating that informative message should be printed. Branch to (9).

    2) The first 35 words from the ATT image in first track of batch are read into the interrogate buffer.

3) The operator number, start of current run and accumulated time are read into 4 temporary locations.

4) If a batch is in ENTRY mode and total number of records is less than 16, the batch is deleted.
The ABT entry is cleared.
Number of entries in the ABT is decreased by one.
The ABT is written back to disk.
The tracks of the batch are released.

A flag is set indicating that an informative message with the job name, batch number and operator number should be printed.
The program branches to the point where job entry is read (9).

5) The accumulated time is updated.

WRTATT routine in PIO saves the accumulated clock in word 34 of ATT. If the last ATT was written by FRZ routines, the accumulated time was already updated. Such ATT's may be recognized by the negative record count. If word 34 is 0, it means this ATT was written back to disk by the CLR program after the accumulated time has been updated.

Updating is done by subtraction of the start time of the current run from the last clock that was saved, and adding this value to the accumulated time.

6) If the batch was in Entry mode and under document control the entry in the LDT is saved in relative mode, as a displacement to the beginning of the document. This is necessary if the batch is put into the frozen state and the LDD is squeezed while the batch is frozen. If the ATT was written by FRZ programs the LDT entry is already saved in relative mode. The document entry is read and the difference between the displacement of current group (ELDT) and the displacement of the beginning of the document is calculated and saved in the ELDT.

7) Word 34 of the ATT image in the interrogate buffer is cleared.

8) The current clock and the updated accumulated time are written to first track of batch.

9) The job entry is read from LJD according to the job index. If flags "job deleted" or "batch's disk is deselected" are set branch to (12). The job name is transferred to the output queue if it is a CRT.

DOCUMENT CLASS _____ CLR _____

PROGRAM NAME _____ Extended Power Fail Recovery (1) _____

10) A check is done that the batch can be reactivated. The following conditions should be met:

10a) The station of the batch was initialized in this run and is now idle.

10b) There is enough memory for the TRB length specified in the job's entry.

If possible a buffer is allocated and linked to the TRB backwards pointer in the ATT of the executing station.

10c) There is enough memory for the format.

The FBA is searched to check if the format is already present in memory. If not, the LFD entry of the format is read and a buffer equal to the length of the format is allocated if possible. The buffer is immediately released.

11) If one of these three conditions is not met the batch is assigned the frozen state. The monitor check override bit in the ATT image in the interrogate buffer is set.

The batch mode in the ABT entry is modified to "frozen".

The ATT image in the interrogate buffer is written on the ATT image in first track of batch.

12) An informative message with the job name, batch number and operator number and the batch status (deleted, frozen) is printed on the teletype.

12) An informative message with the job name and batch number is printed if the disk of the batch is deselected.

13) For the restored batch, the ATT image in the interrogate buffer is written on the ATT image in the first track of batch. Two words have been modified: Word 34, in which the clock was saved, was cleared; the ELDT, word 16, was saved in its relative mode.

14) Second overlay CLR2 is loaded and executed. Its duty is to restore the batch to its station. Meanwhile the CLR overlay is released. When CLR2 completes its operation, it reloads CLR back to memory and execution begins at the second entry point.

Second Entry Point ENT2

d. When all the entries of the ABT have been checked the ABT is written back to disk.

e. If it is the last ABT, care is taken to read into memory the last ABT that is not empty. The message CYBERDATA SYSTEM is displayed on all the idle stations, the message CYBERDATA READY is printed on the teletype.

  System lock is cleared, the interrogate buffer is released, and the program exits to the dispatcher.

Internal subroutines.

ABTIO   reads or writes an Active Batch Table.

WRTRW   reads (writes) the 35 first words of the ATT from (to) the ATT image in first track of batch to (from) the interrogate buffer.

RWSACC   reads (writes) start time of current run, operator number and accumulated time from (to) first track of batch to (from) the 4 temporaries in ATT (ATEMP).

READZ   reads 4 words from disk to 4 temporaries in communication area in ATT of executing station.
        On entry: A - sector address
                   Q - word address
        Used in order to read:
                1. Document entry
                2. Job entry

FRZBCH   sets the monitor check override bit in the ATT image in the interrogate buffer.
        Modifies the batch status in the ABT to frozen.

WRTR   reads (writes) the list of tracks from (to) Tape Recovery Area to (from) 9 temporary, words 50 on, in communication area in ATT of executing station.

INTERFACE

1. ENTRY METHOD

VLSTRT is responsible for loading and executing the CLR program from its first entry point CLR.

```
CLR A
LDQ = N$803     third miscellaneous program
RTJ - (ENTPRG)
```

CLR2 is responsible for loading and executing the CLR program from its second entry point ENT2.

```
ENQ 1           second entry point.
LDQ = N$803
RTJ - (ENTPRG)
```

DOCUMENT CLASS ___CLR_____

PROGRAM NAME_____Extended Power Fail Recovery (1)_____

2. INPUT PARAMETERS OR CONDITIONS

CLR - first entry point assumes I = TTY ATT base address.
ENT2 - second entry point assumes that the following words in communication
       area in ATT of teletype have not been changed while executing CLR2
       program:

EQU SC (46)
SQ (SC + 1) - the address of current ABT entry
ABT (SC + 2) - ABT core address
ABTDA (SC + 8) ABT disk address
TABNO (SC + 9) displacement in sectors of current ABT from the ABT
                disk address.
I register points to the ATT of TTY.

3. EXIT METHOD

a. CLR program exits each time a batch has to be restored to its station and
   CLR2 has to be executed.

RTJ (EXDFUN)
CLR A
LDQ = N$811 CLR2
RTJ - (ENTPRG)

b. When all the ABT's have been scanned CLR exits to the dispatcher.

4. OUTPUT PARAMETERS OR CONDITIONS

a. When a batch is deleted or frozen the following messages:

1.) JJJJJJ, BBB, ENTERED BY 000, IS DELETED

2.) JJJJJJ, BBB ENTERED BY 000, IS FROZEN

are printed respectively

JJJJJJ - job name
BBB   - batch number
000   - operator number

If the disk of the batch is deselected the message
"JJJJJ, BBB IS ON DESELECTED DISK" is printed.

b. Before the CLR program exits to the dispatcher, the message:

xxxxxx CYBERDATA SYSTEM xxxxxx

is displayed on all the idle stations.
The message CYBERDATA READY is printed on the teletype.

3∂6

DOCUMENT CLASS ___CLR___

PROGRAM NAME___ Extended Power Fail Recovery (1)___

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. Low Core Masks

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)

    b. SST

        SLOCK (1)
        ATTB (10)
        CATT (11)
        CLOCKS (17)
        ATTLNG (19)
        STEMP (20)

    c. EST

        LJDDA (8)
        LDDA (9)
        FBA (15)
        ABTADR (22)
        SPIL (23)
        LUND (25)
        LFDDA (39)

    d. ATT

        FTOB (1)
        CMODE (2)
        JOBIND (3)
        RCNT (4)
        TRB (6)
        FMTNO (13)
        BMOD (15)
        ELDT (16)
        DOCNO (17)
        SW2 (20)
        IRB (27)
        RCD (32)
        ATOPT (33)
        ECODE (34)
        BLUN (35)
        ATEMP (73)

327

DOCUMENT CLASS _____ CLR

PROGRAM NAME _____ Extended Power Fail Recovery (1)

   e. Communication area

     SC (36 + 10)
     TRN (SC)   Station number
     SQ (SC + 1)   Address of current ABT entry
     ABT (SC + 2)   ABT core address
     TATT (SC + 4)   ATT address of the batches station
     ABTDA (SC + 8)   ABT disk address
     TABNO (SC + 9)   Displacement in sector of current ABT to ABTDA
     OPR (SC + 10)   The number of the operator who entered the batch
     ENTIND (SC + 12)   FLAG
                    (i) entry or verify mode o/not o
                    (ii) batch on deselected disk, deleted or not
                       deleted. $FFFF negative (not $FFFF)/positive
     TEMPR (SC + 13)   4 temporaries

   f. General equates
     CLRTWO ($0811)   - function index of CLR2.
                    11th miscellaneous program.
     DISPCH ($EA)   - holds dispatcher address
     FW (3)   - FWRITE request code
     FR (2)   - FREAD request code
     EXTBL ($E3)   - Extended system table 4th entry points to the core
                    image disk address
     CORIMG (4)   - 4th entry in the extended system table in core image
                    disk address.

2. MACROS USED
   None

3. PROGRAM DEPENDENCIES

   Entry points

   CLR - called by VLSTRT
   ENT2 - called by CLR2 every time it completes the reactivation of a batch.

   External List

   GETBUF - entry in PIO.
         Gets an interrogate buffer which is used for:
         a) holding the 35 first words of ATT image
         b) holding the messages to be displayed on idle stations or printed
            on the teletype.
   MOVREC - entry in PIO.
         Moves messages to be displayed to the interrogate buffer.
   SETMOV - entry in ENEXSR.
         Enables program to move; used in all internal I/O routines and 2 entry
         points. CLR is released while a batch is restored to its station by
         CLR2 and then reloaded again, thus it does not always reside in the
         beginning of the FRP and should be moveable in order to enable the
         reactivation of as many batches as possible.
   BADTRK - entry in DAM
         address of Bad Tracks Directory.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CLR_____

PROGRAM NAME_____Extended Power Fail Recovery (1)_____

CLRALL	- Entry in DISPLY.
	Clears the screens of all idle stations.
DLINE	- Entry in DISPLAY.
	Displays the line xxxxxxCYBERDATA SYSTEMxxxxxx on all idle
	stations.
RELTRT	- Entry in DAM.
	Releases the tracks listed in the Tape Recovery Area.
LINKTR	- Entry in BORENT.
	Releases all the tracks listed in first track of batch of a deleted
	batch.
SIABTW	- System installation parameter.
	ABT length in words.
SIABTS	- System installation parameter.
	ABT length in sectors.
SITRDA	- System installation parameter.
	Tape Recovery Area disk address.

Routines used by means of transfer vector

CLCDWA ($90) - Entry in PIO.
              Calculate disk word addresses for the internal I/O routines:
              WRTRW
              RWSACC
              READ2
              WRTR

REDDWA ($A9) - Entry in PIO.
              Reads from disk with disk word addressing used in the internal
              I/O routines:
              WRTRW
              RWSACC
              READ2
              WRTR

WRTDWA ($AA) - Entry in PIO.
              Writes to disk with disk word addressing used in the internal
              I/O routines:
              WRTRW
              RWSACC
              WRTR

REDSEC ($A7) - Entry in PIO.
              Reads from disk with sector addressing.
              Used in the internal I/O routine
               ABTIO

3.24

DOCUMENT CLASS ____CLR____

PROGRAM NAME____ Extended Power Fail Recovery (1) ____

WRTSEC ($A8) - Entry in PIO.
                    Writes to disk with disk sector addressing.
                    Used in the internal I/O routine
                    ABTIO

ENTPRG ($9E) - Entry in ENTPRG.
                    Used in order to load CLR2 overlay whenever a batch should
                    be restored to a station.

EXDFUN ($9F) - Entry in ENTPRG.
                    Used by CLR in order to release its area in memory before
                    loading CLR2 and on completion.

CAM ($AE)   - Entry in CAM.
                    Used in order to allocate memory for the TRB of an active
                    batch. Allocates a buffer for the format of an active batch
                    which is released later in order to make sure that there is
                    enough memory to reactivate the batch.

RELFOR ($AD) - Entry in GFCORE.
                    Releases the interrogate buffer when program completes.
                    Releases the TRB in case there is not enough memory for the
                    format of an active batch.

SUPRW ($97) - Entry in PIO.
                    Used in order to print informative messages a, b, c and the
                    message CYBERDATA READY on the teletype.

DISPLY ($A0) - Entry in DISPLY.
                    The display driver.

ENTSR ($91) - Entry in ENEXSR.
                    Saves return address in enter/exit stack for the internal
                    I/O routines:
                    WRTR
                    WRTRW
                    RWSACC
                    READ2
                    ABTIO

EXTSR ($92) - Entry in ENEXSR.
                    Restores the return address for the internal I/O routines:
                    WRTR
                    WRTRW
                    RWSACC
                    READ2
                    ABTIO

330

DOCUMENT CLASS _____CLR_____

PROGRAM NAME_____Extended Power Fail Recovery (1)_____

4. DATA AREAS AND SYSTEM TABLES USED

SST
ATT
ABT - on disk and in core
LJD
LDD
LFD
FBA
Interrogate buffer
DTL - (i) ATT image in first track of batch.
      (ii) Operators number, start time of run and accumulated time in first
           track of batch.
ADTOPT - job name is moved to the header if station is a CRT.
BADTRK

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

Program size - 650 words.
Interrogate buffer - 39 words long.

2. RELOCATION TYPE

Disk resident.
Run anywhere.

3. ENTRANCY CLASSIFICATION

Not reentrant.

DOCUMENT CLASS ___CLR2___

PROGRAM NAME___Extended Power Fail Recovery (2)___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

CLR2 restores a batch to its station. The ATT of the station is updated with the ATT image and the screen is regenerated. CLR2 program uses parameters in the ATT of the teletype which were prepared by CLR program, and the ATT image from first track of batch in the interrogate buffer of the teletype. The ATT image was read by CLR program.

CLR2 recognizes an ATT image that was written by a FRZ program according the negative record count.

2. DETAILED DESCRIPTION

a. Function lock is set on the ATT of TTY in order to stop processing characters which are entered on TTY while modifying another ATT for the reactivation of the batch.

b. The following words are transferred from the ATT of TTY to the ATT of the station of the batch.

(i) Logical unit.

(ii) Function index (function index of CLR2 program). Function index is needed to calculate the return address according to the relative address in the enter/exit stack.

c. The ATT image in the interrogate buffer of TTY is transferred to the ATT of the station.

First 32 words are transferred except for the function index.

d. If the ATT was written by FRZ programs the error locks are restored FRZ saves them in the PFBA.

e. The TRB is disconnected from the ATT of TTY and is linked to the ATT of the station.

f. The format of the batch is brought into memory.

g. The monitor check override bit in switch 2 is set in order to avoid checking while modifying the ATT.

h. Words 34-64 are read from the ATT in first track of batch into the ATT in core.

i. If a batch is in ENTRY mode and under document control the "entry in LDT" is modified from relative to absolute form.

The document's entry from LDD is read and the displacement of current group to beginning of LDD is calculated as the sum of the displacements of first group (taken from LDD) and of current group to first group (current ELDT) and updates ELDT.

3IC

j. The last processed record is read into the TRB. The disk address of the record is taken from the current disk address in ATT.

k. End of Data state is restored.
The state is recognized according to the following flags:

   (i)   Where the ATT was written by CORKEY in such state, i.e. a record was inserted after the last one or the last record was deleted, bit 15 of word 5 of the ATT is set.

   (ii) Where the ATT was written by FRZ programs in such state, bit 15 in the interrogate buffer backwards printer is set.

   (iii) If it is End of Data in verify the "no forward motion" bit in switch 2 is set.

   The current disk address from ATT is stored in previous record disk address of TRB.

k2. The "format change bit" is restored according bit 13 in AINT, if ATT was written by FRZ.

l. Current disk address is restored, if ATT is of FRZ. This disk address was saved in the interrogate and search buffers backwards pointers by FRZ programs.

m. The data connected with the format is regenerated.

   If last ATT was written by FRZ, the format might have been changed while the batch was frozen.

   The field number, relative format field address and character count in current field are regenerated.

   The End of Field state is restored if a flag $FF was left in the field count by a FRZ program.

n. Data type is set and the screen is regenerated.

o. The monitor check override bit and the 'LL' bit are cleared.

p. The entry point that should be executed next is stored in the enter/exit stack.

   If the station must wait for the next char. or key to be entered no address is saved.

   This happens if the batch is in ENTRY mode or READ mode. If the batch is in VERIFY mode, no verification options are specified, no special key has to be entered and the record is not an inserted one, the field count and record count are updated with the values of the "end of field" and EORTST, entry point in VMM, is stored in the enter/exit stack.

   If the batch is in VERIFY mode at the beginning of a record, VMM1, the check field entry point in VMM is stored in the enter/exit stack.

DOCUMENT CLASS ___CLR2___

PROGRAM NAME___Extended Power Fail Recovery (2)___

If in VERIFY and the conditions listed above are not fulfilled no address is saved.

If an address is saved the pseudo-function lock is set on the ATT, thus MPC will process this station from the address in the enter/ exit stack.

q. The function lock of the ATT of TTY is cleared, the CLR2 program is released and the CLR program is reloaded and executed from the second entry point.

INTERFACE

1. ENTRY METHOD

   CLR A   First entry point
   LDQ = N$811   11th miscellaneous program
   RTJ - (ENTPRG)   load and go

2. INPUT PARAMETERS OR CONDITIONS

   CLR loads and executes the CLR2 program whenever a batch should be restored to a station and there is enough memory for the TRB and the format.

   (i)   The following parameters are transferred via the communication area of the teletype's ATT/
           EQU   SC(36+10)
           TATT   (SC+4) - address of the batch's ATT.

   (ii)   The TRB for the batch of the length specified in the job entry is linked to the TRB backwards pointer in the teletype's ATT.

   (iii)   The logical unit of the batch is in the appropriate word in the teletype's ATT.

   (iv)   The 35 words of the ATT image from first track of batch are in the interrogate buffer of the teletype's ATT.

   (v)   I register points to the ATT of TTY.

3. EXIT METHOD

   Return to caller. Reload the CLR program and re-execute it from the instruction following the load and execute of the CLR2 program (second entry point).

4. OUTPUT PARAMETERS OR CONDITIONS

   I register points to the ATT of TTY.

DOCUMENT CLASS ___CLR2_____

PROGRAM NAME_____Extended Power Fail Recovery (2)_____

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. Low Core Masks

      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)

   b. SST

      SST ($47)
      EXTSST (0)
      SLOCK (1)
      BATT (10)
      CATT (11)
      CFFA (12)
      IBL (14)
      TIQ (15)

   c. EST

      LDDDA (3)

   d. ATT

      FTOB (1)
      CMODE (2)
      FCNT (3)
      RCNT (4)
      ASCNT (5)
      TRB (6)
      PFBA (7)
      CURDA (9)
      FMTNO (13)
      RFA (14)
      BMODE (15)
      ELDT (16)
      DOCNO (17)
      SW1 (19)
      SW2 (20)
      FUNIDX (21)
      AINT (27)
      ASRCH (28)
      LUN (35)
      ATEMP (73)

313

DOCUMENT CLASS ___CLR2_____

PROGRAM NAME_____Extended Power Fail Recovery (2)_____

    e. Communication Area

        SC (36 + 10)
        TATT (SC + 4) - address of ATT of batch's station

    f. General Equates

        CLR1 ($803) - function index of the CLR program.

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   Entry points

   CLR2

   External list

SETMOV	- Entry point in ENEXSR. Enables program to move while doing I/O or waiting for core.
SETDIS	- Entry in DISPLY. Sets display to DATA mode.
REGEN	- Entry in DISPLY. Regenerates the screen when reactivating a batch.
STTYPE	- Entry in EMM. Sets data type according to the format.
EORTST	- Entry in VMM. This address is saved in the enter/exit stack if this entry should be executed in the next cycle.
VMM1	- Entry in VMM. This address is saved in the enter/exit stack if this entry should be executed in the next cycle.

   Routines called via transfer vector

GFCORE ($9D)	- Entry in GFCORE. Gets the format for the batch.
ENTSR ($91)	- Entry in ENEXSR. Saves the address of the entry points in VMM that should be executed in the next cycle in the enter/exit stack.
READN ($A6)	- Entry in PIO. Reads the last processed record into the TRB.
CLCDWA ($90)	- Entry in PIO. Calculates the disk word addresses of: 1. document's entry 2. counters area in ATT image in first track of tach.

314

DOCUMENT CLASS ___CLR2_____

PROGRAM NAME_____Extended Power Fail Recovery (2)_____

REDDWA ($A9) - Entry in PIO.
      Reads according to disk word addressing:
      1. counters area from ATT image in first track of batch to
        the counters area in ATT in core.
      2. document's entry from LDD to a temporary in ATT (ATEMP + 1).
EXDFUN ($9F) - Entry in ENTPRG.
      Used in order to release the area CLR2 occupies in memory.
ENTPRG ($3E) - Entry in ENTPRG.
      Reloads CLR program and jumps to its second entry point.
DISP ($A0) - the display driver.

4. DATA AREAS AND SYSTEM TABLES USED

SST
ATT
Interrogate buffer
TRB
LDD
DTL-(i) ATT image

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

Program size - $11D_{16}$ = 285 words.

2. RELOCATION TYPE

Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

315

DOCUMENT CLASS CYBERDATA

PROGRAM NAME   CMISCI - CRT Miscellaneous Routines   Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    CMISCI has five entry points:

    a. CDIS sets the screen area data definition bits in TOPT.

    b. CLOK sets and threads, if necessary, the system lock flags in TOPT.

    c. ENDSR saves the return address for display routines in the PIO area
        of the ATT.

    d. EXDSR returns control to the last address saved via ENDSR.

    e. CDIR sets up the parameters for a direct call to CDATA.

2. DETAILED DESCRIPTION

    a. CIDS sets TOPT/BLOKS/13-12 according to value passed in A and tempo-
        rarily saved in ATT/RQ. Values are: 00 = Data
                                               01 = Interrogate
                                               10 = Idle

    b. CLOK checks to see if system locks are set. If so, it exits via EXTSR.
        If locks were not set, it checks to see if its TOPT is threaded for
        output. If the TOPT is threaded, the locks field is set and exit is
        via EXTSR. If the TOPT was not threaded, CLOK threads it and sets TOPT/
        BLOKS/4.6 (system locks) and exit via EXTSR. Interrupts are inhibited
        during all threading and locks setting.

    c. ENDSR saves return addresses of display routines using ATT/PIO area as
        a stack because the display routines do not use PIO routines. ENDSR
        checks for stack overflow and <u>hangs</u> if overflow occurs.

    d. EXDSR exits using the last return address stored by ENDSR, and decreases
        the stack pointer by 1.

    e. CDIR calls ENDSR to save its return address. A (saved in ATT/RQ) points
        to the five-word parameter block used for a call to CDATA. The block
        is moved internally to CDIR where a call to CDATA is made. Exit is via
        EXDSR.

INTERFACE

1. ENTRY METHOD

    CDIS, CLOK, ENDSR, CDJR are entered by return jump. EXDSR is entered by
    direct jump.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME __CMISCl - CRT Miscellaneous Routines__ __Level 06__

2. INPUT PARAMETERS OR CONDITIONS

CDIS:

I = current ATT base address.

CLOK:

I = current ATT base address.

ENDSR:

I = current ATT base address.

EXDSR:

None

CDIR:

I = current ATT base address

ATT/RQ = pointer to 5-word parameter block for CDATA call.

3. EXIT METHOD

CDIS: Return to caller

CLOK: JMP- (EXTSR)

ENDSR: Return to caller

EXDSR: Jump to last address stored in display routines exit stack.

CDIR: Return via EXDSR

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      ZERO   ($22)

      SST   ($47)

      CDATA ($80)

      EXTSR ($92)

   b. ATT

      TOPT   (33)

      RQ   (65)

c. SST

TOPCRT (4)

TEMP   (20)

d. TOPT

BLOKS   (14)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

ENEXSR

CDATA

4. DATA AREAS USED

SST

ATT

TOPT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$3F_{16} = 63_{10}$ words.

2. RELOCATION TYPE

Loader relocatable.

3. ENTRANCY CLASSIFICATION

Dormant reentrant.

DOCUMENT CLASS __CONCAN__

PROGRAM NAME_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    CONCAN checks for a continue or cancel request at the end of a display to
    the operator.

2. DETAILED DESCRIPTION

    CONCAN calls the common entry subroutine to save the return address in the
    ATT exits stack. Then the entry point GETSTR in PIO is called to input
    one character into the interrogate request buffer. If that character was
    a continue key code, A is cleared and exit is made via the common exit
    subroutine. If a continue key code was not input, the completion code in
    word 2 (bits 8 -15) of the interrogate request buffer or ATT/TRMNT/14 is
    checked for a 1 which indicates that the cancel key was entered. If so, A
    is set to 1 and exit is via the common exit subroutine.

    If neither a continue or cancel was entered, the one character input is
    requested again and again until one of those keys is entered.

INTERFACE

1. ENTRY METHOD

    Return jump

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address

3. EXIT METHOD

    Return to caller after continue or cancel is entered by the operator.

4. OUTPUT PARAMETERS OR CONDITIONS

    I = current ATT base address

    A = 0 for continue

    A ≠ 0 for cancel

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK ($2)
        ENTSR ($91)
        EXTSR ($92)
        GETSTR ($A2)

DOCUMENT CLASS __CONCAN__

PROGRAM NAME_____

      b. ATT

         TRMNT (23)
         AINT (27)

      c. CONSTANT

         CONT ($75)   Keyboard code for continue key

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    ENEXSR
    PIO

4. DATA AREA USED

    ATT
    INT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $15_{16} = 21_{10}$ words.

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Reentrant.

161

PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME_____ CON029 - 029 Keyboard Character Conversion   Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    CON029 converts keyboard with/without adding machine input codes to
    ASCII equivalents.

2. DETAILED DESCRIPTION

    If the terminal is error locked, only 'Rest' and 'Display' keys are allowed.
    If an unused key (with code $17) was entered, the code is passed on only if
    the terminal was in interrogate or idle mode. Otherwise, a 'PP' is displayed
    at the terminal. If the code entered is alpha or a lower case key that is not
    a function key and the field data type is defined as lower case, no code
    conversion is necessary. If the field data type is defined as any form of
    numeric (upper case), a table lookup is performed to convert the entered
    code to its ASCII equivalent. Special checks are performed to ignore
    certain codes ($17) not used on a 029 keyboard. If one of these codes is
    found, exit is to the dispatcher (the code is effectively ignored).

INTERFACE

1. ENTRY METHOD

    RTJ CONO 029 keyboard type

              or

    RTJ CONADD 029 + adding machine keyboard type.

2. INPUT PARAMETERS OR CONDITIONS

    I = ATT base address. ATT/ATTMP = keyboard code for key entered.

3. EXIT METHOD

    P + 1 return to caller if key is processed. Jump to dispatcher if terminal
    error locked or key unused. Jump to ERROR if PARITY ERROR.

4. OUTPUT PARAMETERS OR CONDITIONS

    I = ATT base address. ATT/ATTTMP = keyboard code for key entered.
    ATT/CURCH/7-0 = ASCII character or 0 (for function). When key entered was
    a function. Q = 0, otherwise, Q = ASCII value of the character entered.
    Q = $1A (code of PP) when exit is to ERROR.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ONEBIT ($23)
        ERROR ($96)
        DISPCH ($EA)

    b. ATT

        CMODE (2)
        CURCH (2)
        DT (13)
        ATTTMP (73)

    c. CONSTANTS

        CPP ($1A)   "PP" error code
        RES ($67)   Reset key code
        DSPKEY ($73)   Display error key code

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

    NDISP
    GSE
    DISPCH
    ERROR

4. DATA AREAS USED

    Internal Table MTA which contains the 029 conversions for upper case key
    codes.

    ATT

PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME_____ CON029 - 029 Keyboard Character Conversion _____ Level 06

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $88_{16} = 136_{10}$ words.

2. RELOCATION TYPE

   Loader relocatable.

3. ENTRANCY CLASSIFICATION

   Not reentrant.

ATTACH ANALYTICAL SUPPLEMENT

Attachment A contains key codes and their corresponding 029 and TTY characters.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___ CON029 - 029 Keyboard Character Conversion _____ Level 06

The hardware for the adding machine is not
available as a standard product.

DOCUMENT CLASS CYBERDATA

PROGRAM NAME   CON029 - 029 Keyboard Character Conversion   Level 06

ATTACHMENT A

Key Code	029 Character	TTY Character
00		
01		
02		
03		
04		
05		
06		
07		
08		
09	Not Used	Not Used
0A		
0B		
0C		
0D		
0E		
0F		
10		
11		
12		
13		
14		
15		
16	0 (adding machine)	
17	.	%
18	,	#
19	-	*
1A	BSC	6
1B	(Not used)	[
1C	DUP	7
1D	(Not used)	&
1E	(Not used)	1
1F	(Not used)	!
20	Space	Space
21	!	<
22	"	C
23	#	@
24	$	$
25	%	3
26	&	P
27	'	"
28	(	=
29	)	E
2A	*	4
2B	+	Q

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME __CON029 - 029 Keyboard Character Conversion__   Level 06

Key Code	029 Character	TTY Character
2C	,	;
2D	.	8
2E	·	·
2F	/	9
30	0	(
31	1 (7 on adding machine)	\
32	2 (8 on adding machine)	!
33	3 (9 on adding machine)	0
34	4	;
35	5	K
36	6	L
37	7 (1 on adding machine)	[
38	8 (2 on adding machine)	·
39	9 (3 on adding machine)	·
3A	:	D
3B	;	F
3C	<	5
3D	=	>
3E	>	S
3F	?	X
40	@	2
41	A	A
42	B	B
43	C	C
44	D	D
45	E	E
46	F	F
47	G	G
48	H	H
49	I	I
4A	J	J
4B	K	K
4C	L	L
4D	M	M
4E	N	N
4F	O	O
50	P	P
51	Q	Q
52	R	R
53	S	S
54	T	T

DOCUMENT CLASS _____ CYBERDATA

PROGRAM NAME_____ CON029 - 029 Keyboard Character Conversion _____ Level 06

Key Code	029 Character	TTY Character
55	U	U
56	V	V
57	W	W
58	X	X
59	Y	Y
5A	Z	Z
5B	[	'
5C	\	T
5D	]	R
5E	∧	+
5F	—	W
60	(Not used)	/
61	(Not used)	?
62	(Not used)	DUP
63	(Not used)	REL
64	(Not used)	BSC
65	ESB	ESB
66	00 (adding machine)	(Not used)
67	RESET	RESET
68	SKIP	SKIP
69	BSR	BSR
6A	BOR	BOR
6B	(Not used)	FMT BCK
6C	(Not used)	FMT ADV
6D	FMT BCK	)
6E	FMT ADV	0
6F	REL	–
70	DOC BCK	DOC BCK
71	DOC ADV	DOC ADV
72	BSF	BSF
73	DISP	DISP
74	REP FLD	REP FLD
75	CONT	CONT
76	CNCL	CNCL
77	MLTL DUP/SKIP	MLTL DUP/SKIP
78	INS	INS
79	DLT	DLT
7A	ERR OVR	ERROUR
7B	FORM SEL	FORM SEL
7C	COR	COR
7D	RTRN	RTRN
7E	READ	READ
7F	INT	INT

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___CONT_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   CONT processes the CONT key.

2. DETAILED DESCRIPTION

   CONT checks if switch 1 bit 5 is set for 'CONT must be next'. If it is
   not, ERROR is called with error code 4. Else character count in current
   field, and character count in record will be increased to point to end
   of field. IDK bit in CMODE and bit 5 in switch 1 are cleared. CONT
   exits to ENDOF1 in VMM for end of field processing.

INTERFACE

1. ENTRY METHOD

   CONT is loaded by IDK via ENTPRG, when CONT key is entered, and terminal
   is in IDK mode.

2. INPUT PARAMETERS OR CONDITIONS

   I = current ATT base address.
   Switch 1 bit 5 should be set.
   IDK mode bit is used.

3. EXIT METHOD

   JMP - (ERROR) - with error code - 4, when 'CONT must be next' bit is not set.
   JMP + ENDOF1 - Entry point in VMM.

4. OUTPUT PARAMETERS OR CONDITIONS

   I = current ATT base address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)
      SST ($47)
      ERROR ($96)

    b. GENERAL

       FW ($3)
       B ($42)

    c. SST, ATT, EST

       CATT (11)
       SST ($47)
       SLOCK (1)
       STEMP (20)
       ATTB (10) ABT ADR (22)
       FTOB (1)
       JINX (3)
       ABTINX (22)
       TRMN (23)
       AINT (27)
       DSKST (29)
       LUN (35)

    d. SCA

       SUPCOM (36)
       SC (SUPCOM)
       FLAG (SC+10+0)
       ATEMP1 (SC+10+1)
       MTHREE (SC+10+2)
       MTWO (SC+10+3)
       BBB (SC+10+4)
       CCC (SC+10+5)
       BBC (SC+10+6)
       TEMP (SC+10+7)
       JOBN (SC+10+9)
       JOBNAM (SC+10+10)
       TBATCH (SC+10+13)

2. INTERNAL ROUTINES

   CHECK4 - To check if actual batch number is less than the last
         batch number (BBC-CCC)
   In case condition does not hold return is made to sup.

3. MACROS USED

   None.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____CONT_____ Level 6

INTERFACE

1. ENTRY METHOD

   DBT is loaded by ENTPRG. Execution begins at only entry point DBT.

2. INPUT PARAMETERS (in the DBT command)

   Job name
   Initial Batch number - required
   Last Batch number
   ,B - optional

3. EXIT METHOD

   a. Completion exit

      JMP - (SV)

   b. Cancel exit

      ENA 39
      JMP - (REJSOP)

   c. Error exit

      ENA 0 (3,8)
      JMP - (REJSUP)

4. OUTPUT CONDITIONS

   I = current ATT address

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK ($2)
      NZERO ($12)
      ONEBIT ($23)
      ZROBIT ($33)
      ZERO ($22)
      WAIT ($B0)
      REJSUP ($B2)
      SV ($B1)
      GETBIN ($99)
      SPILL ($A1)
      GETALP ($9B)
      GETCOM ($9C)
      SUPRW ($97)

    b. SST

        CFFA (12)

    c. ATT

        MODE (2)
        FCNT (3)
        RCNT (4)
        SW1 (19)

    d. CONSTANTS

        CA4 (4)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

    ENDOF1 - Entry point in VMM
    ERROR   - F or displaying error codes.

4. DATA AREAS USED

    ATT
    SST
    FMT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $(1F)_{16} = (31)_{10}$.

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

4. PROGRAM DEPENDENCIES

    SUPRW   -
    REJSUP   -
    WAIT   -
    GETBIN   -
    SV   -
    SPILL   -
    GETALP   -
    GETCOM   -
    BD   -
    DB   -
    GETLJD   -
    SETMOV   -
    MOVREC   -

5. DATA AREAS AND SYSTEM TABLES USED

    SST
    EST
    ATT
    LJD
    ABT
    SCA
    INT. BUFFER

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $101_{16} = 256$ words of program.

2. RELOCATION TYPE

    Run anywhere, disk resident.

3. REENTRY

AA5641

PRINTED IN THE USA

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___CONTTY___

PROGRAM NAME___Teletype Keyboard Code Conversion___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    CONTTY converts TTY keyboard input codes to ASCII equivalents. (No
    consideration is given to field data type, e.g. upper shift.)

2. DETAILED DESCRIPTION

    If the terminal is a TTY, the conversion table is plugged for code $66 (a
    code not used on the 029 keyboard). If the code entered was less than
    $16 and the terminal was not error locked, a "PP" error is displayed at
    the terminal. (If the terminal was already error locked, exit is to the
    dispatcher--the key is ignored.) If the code entered was $16 or greater,
    a table lookup is performed to get the ASCII equivalent. A zero table
    entry means the key is not used and exit is to the dispatcher (the key is
    ignored). The converted character is stored in ATT/CURCH/7-0 and Q is
    cleared if the key entered was a function key. If the terminal was error
    locked, reset or display keys are processed and all others are ignored.

INTERFACE

1. ENTRY METHOD

    Return jump

2. INPUT PARAMETERS OR CONDITIONS

    I = ATT base address

    ATT/ATEMP = keyboard code for key entered

3. EXIT METHOD

    P + 1 return to caller if key is processed. Jump to dispatcher if terminal
    error locked or key unused.

4. OUTPUT PARAMETERS OR CONDITIONS

    I = ATT base address

    ATT/ATEMP = keyboard code for key entered

    ATT/CURCH/7-0 = ASCII character or unconverted code for functions

    Q = 0 when key entered was a function

    Q ≠ 0 when key entered was a character

DOCUMENT CLASS ___CONTTY_____

PROGRAM NAME___ Teletype Keyboard Code Conversion _____

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

       LPMASK (2)
       NZERO ($12)
       ONEBIT ($23)
       SST ($47)
       ERROR ($96)
       DISPCH ($EA)

    b. ATT

       CMODE (2)
       CURCH (2)
       ATEMP (73)

    c. SST

       STEMP (20)

    d. CONSTANTS

       CPP ($1A)   "PP" error code
       RES ($67)   Reset key code
       DSPKEY ($73)   Disp key code

2. MACROS USED

   None

3. PROGRAM DEPENDENCIES

   NDISP
   GSE

4. DATA AREAS USED

   ATT
   SST
   Internal table MTA which contains the TTY conversion codes, stored two characters per word.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $6B_{16} = 107_{10}$ words

DOCUMENT CLASS <u>CONTTY</u>

PROGRAM NAME <u>Teletype Keyboard Code Conversion</u>

2. RELOCATION TYPE

Loader relocatable

3. ENTRANCY CLASSIFICATION

Dormant reentrant

ATTACH ANALYTICAL SUPPLEMENTS

Attachment A contains key codes and their corresponding 029 and TTY characters.

TELETYPE/TYPEWRITER CONFIGURATION

80

DOCUMENT CLASS CONTTY

PROGRAM NAME Teletype Keyboard Code Conversion

ATTACHMENT A

Key Code	029 Character	TTY Character
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
0A		
0B	Not Used	Not Used
0C		
0D		
0E		
0F		
10		
11		
12		
13		
14		
15		
16		
17	.	%
18	,	#
19	–	*
1A	BSC	6
1B	(Not used)	{
1C	DUP	7
1D	(Not used)	&
1E	(Not used)	]
1F	(Not used)	}
20	Space	Space
21	!	<
22	"	C
23	#	@
24	$	$
25	%	3
26	&	P
27	'	"
28	(	=
29	)	E
2A	*	4
2B	+	Q

81

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CONTTY__

PROGRAM NAME __Teletype Keyboard Code Conversion__

Key Code	029 Character	TTY Character
2C	,	,
2D	-	-
2E	.	.
2F	/	9
30	0	(
31	1	>
32	2	I
33	3	0
34	4	;
35	5	K
36	6	L
37	7	Λ
38	8	:
39	9	.
3A	:	D
3B	;	F
3C	<	5
3D	=	\
3E	>	S
3F	?	X
40	@	2
41	A	A
42	B	B
43	C	C
44	D	D
45	E	E
46	F	F
47	G	G
48	H	H
49	I	I
4A	J	J
4B	K	K
4C	L	L
4D	M	M
4E	N	N
4F	O	O
50	P	P
51	Q	Q
52	R	R
53	S	S
54	T	T

82

DOCUMENT CLASS ___CONTTY_____

PROGRAM NAME___Teletype Keyboard Code Conversion_____

Key Code	029 Character	TTY Character
55	U	U
56	V	V
57	W	W
58	X	X
59	Y	Y
5A	Z	Z
5B	⌐	,
5C	\	T
5D	¢	R
5E	¦	+
5F	–	W
60	(Not used)	/
61	(Not used)	?
62	(Not used)	DUP
63	(Not used)	REL
64	(Not used)	BSC
65	ESB	ESB
66	(Not used)	00 (TTN only)
67	RESET	RESET
68	SKIP	SKIP
69	BSR	BSR
6A	BOR	BOR
6B	(Not used)	FMT BCK
6C	(Not used)	FMT ADV
6D	FMT BCK	)
6E	FMT ADV	0
6F	REL	–
70	DOC BCK	DOC BCK
71	DOC ADV	DOC ADV
72	BSF	BSF
73	DISP	DISP
74	REP FLD	REP FLD
75	CONT	CONT
76	CNCL	CANCEL
77	MLTL DUP/SKIP	MLDS
78	INS	INS
79	DLT	DEL
7A	ERR OVR	ERROVR
7B	FORM SEL	FSL
7C	COR	COR
7D	RTRN	RETURN
7E	READ	READ
7F	INT	INT

83

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The routine processes the CORRECT key in READ and VERIFY modes.

    In VERIFY mode after a mismatch, places the new character in the TRB, clears
    error flags as necessary, and sets revalidate bit. In READ and PSEUDO ENTRY,
    after replacing a field, clear flags set rewrite, check for reverification
    and autodup.

    In READ it also processes the insertion and deletion of a record, it updates
    pointers to next and previous records in batch, and updates counters and
    error flags (if needed).

2. DETAILED DESCRIPTION

    CORKEY starts execution by checking if SW1 B11 is set, for COR must be next.
    If so, IDK bit is cleared and CORKEY jumps to different sections to handle
    the cases of COR after INSERT, and of COR after DELETE or after replace
    field.

    When replacing a field in READ mode and field is an autodup field and MLDS
    is not set (SW1B6), checks basic mode if it is entry and the record changed
    is not the last entered, or if basic mode is VERIFY and this is not the last
    record of the batch, first word of next record is read into core, bit 13 -
    for format change is set, and the word is written back into core. When in
    READ mode and basic mode is VERIFY it checks if current verify point was
    passed in read. If not, CORKEY will call CHKVRY for checking if reverification
    needed. If it is, mode is changed to pseudo verify, and REWRITE bit in SW1
    is set.   After clearing 'M' from display and calling EXDFUN, CORKEY will exit
    to RVFVER in VMM. If current Verify point was passed or no reverification
    was needed, rewrite bit in SW1 is set, and COR bit is cleared. The internal
    subroutine REVERY is called to set SW2 bit 10 (in Verify and in Read when
    MLDS was set) and SW2 bit 14 (for record reverification). Pseudo entry bit
    is cleared in CMODE - and CORKEY exits to ENDOF1 in VMM (after calling EXDFUN).
    If in Read and the current field is an autodup, set Auto Change bit in header
    if MLDS (SW1B6) was set, and Format Change bit otherwise.

The procedure SETBEG in VMM is called to set pointers back to start of field. DISPLAY is called and CORKEY exits via CLRSR. If SW1 - bit 11 was not set and basic mode is entry, or it is verify without MISMATCH character CORKEY will exit to ERROR, with error code 4. If basic mode is verify and there is a mismatch character, rewrite bit in SW1 is set, and REVERY is called to set forced reverification flag- and Auto dup flag (SW2B10) as required.

If SW2 bit 13 is not set, it is set (for Revalidate), and CORKEY exits to REPKEY for updating counters and flags. It will return from REPKEY through second entry point of CORKEY - COR2. If SW2 bit 13 is already set, control will be passed to COR2 without exiting to REPKEY. Here the new character is placed in the ATT for current character, and it is stored via SBYTE in the TRB. The mismatch word in the ATT is cleared, and CORKEY exits to GETNZ in VMM.

COR after DELETE:

In READ mode: If basic mode is ENTRY, checks if the deleted record is the last in batch and releases free area if possible (via AVAIL). If the deleted record is the last logical record, internal procedure BSRCD is called (see description of the procedure attached). Document and format change bits are transferred from the deleted record to SW1. Bit 15 is set to indicate that the next record will start with a new format; Bit 14 in SW1 is set if a new document should be started in next record. Pointers to previous record, and next disk address are updated, and ATT is written to disk via WRTATT, with end of data bit set. CORKEY exits to SETNFM in PIO. If basic mode in VERIFY or if it is Entry but the deleted record is not the last in batch (logically), BSRCD is called. Relevant information about the record is stored. Available storage is freed, pointers to previous record are updated to point over deleted record, basic mode disk addresses are updated to point to actual next free are released by the delete. Pointers to the record after the deleted one are updated too. First in format and first in document pointers are set in the header of this record. If mode is READ, CORKEY exits to second entry point in BSR, for display. If mode is verify and the deleted record is the first in batch, CORKEY will exit to EORVRC; else, to EORVR1 (both after calling EXDFUN), both entry points in EORVR.

COR after INSERT:

If basic mode is Entry and current record is verifiable, increase number of verifiable records in batch. Increase number of records in READ and number of records in batch. Update basic mode information and pointers according to new record. If it was an INSERT at end of data set last record flag. Update available number of words in track. Check if there was format change, for the inserted record, and set flag for new format in the header of next record. Put free disk address into next, set pointers to beginning of record, and change formats if required. Write ATT to disk, and exit to CLRSR after displaying the record following the inserted one.

Internal Subroutines:

REVERY: Sets SW2 bit 10 in Verify and in Read when NLTL DUP SKIP was set, and the change is done for a record in basic mode. It also sets forced reverification bit if needed.

AVAIL: Checks for free areas and releases them if possible.

BSRCD: Moves to end of record. Calls GETPFD to perform counters. Updates RCD, RCDRD, number of verify records, and flagged records. It is used when processing COR after DELETE.

INTERFACE

1. ENTRY METHOD

   CORKEY is a disk resident program invoked by ENTPRG, when COR key is entered. It has two entry points: CORKEY which is entered when COR key is entered; COR2 is entered when returning from REPKEY.

2. INPUT PARAMETERS

   I = current ATT base address.

3. OUTPUT PARAMETERS

   I = current ATT base address.

4. EXIT METHOD

   a. JMP GETNZ in VMM - when there is a mismatch character, in Verify mode.

   b. JMP RVFVER in VMM, when COR after REPKEY, in Read with basic mode =
       Verify, when Verify point was passed in read.

   c. JMP END OF 1 in VMM - when Verify point was not passed in read.

   d. JMP - (CLRSR) - When processing COR after INSERT, and when replacing
                        field in Read mode.

   e. JMP SETNFM - In PIO, when COR after DEL in READ, with basic mode ENTRY,
                     and at end of data.

   f. JMP EORVRC - In EORVR, when COR after DEL in Verify (for first record).

   g. JMP EORVR1 - In EORVR, when processing COR after DELETE in Verify
                     mode, when processing COR after DELETE in Verify mode,
                     if deleted record was not first in batch.

   h. LDQ =N$1869 exit to second entry point in BSR.
       RTJ- (ENTPRG)
       - When processing COR after DELETE in READ mode, with basic mode - Verify.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. TRANSFER VECTOR

       REDDWA ($A9)
       WRTDWA ($AA)
       WRITEC ($A3)
       READP ($A5)
       REWRIT ($A4)
       READN ($A6)
       WRTATT ($AC)
       GBYTE ($94)
       SBYTE ($95)
       ERROR ($96)
       EXDFUN ($9F)
       GFCORE ($9D)
       ENTPRG ($9E)
       DISPLY ($A0)
       CLRSR ($93)

    b. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)

    c. ATT

        CMODE (2)
        FCNT (3)
        RCNT (4)
        MISCH (5)
        PFBA (7)
        SW1 (19)
        SW2 (20)
        VRCOR (16)
        ATEMP (73)
        CURCH (2)
        BMOD (15)
        ATRB (6)
        BRCNT (15)
        SWITCH (24)
        RCDER (30)
        VRCD (29)
        CURDA (9)
        NDA (11)
        BMDA (17)
        LUNIT (35)
        RQ (65)
        FMTNO (13)
        BFMTNO (25)
        RCSIZE (69)
        RCD (32)
        RCDRD (31)
        AVLTK (8)
        RFA (14)

    d. SST

        STEMP (20)
        CFFA (12)

    e. CONSTANTS

        S (S)
        MMA (6)
        LU (3)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

VMM	- GETNZ
	ENDOF1
DISPLY	- SETBEG
	DBSF
	REGEN
EMM	- STTYPE
EORVR	- EORVRC
	EORVR1
PIO	- REDDWA-
	WRTDWA-
	WRITEC-
	READP-
	REWRIT-
	READN-
	WRTATT-
	SETNFM
GSE	- GBYTE
	SBYTE
	ERROR
EXDFUN	
ENTPRG	
GFCORE	
ENEXSR	- CLRSR-
REPKEY	

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME ___CORKEY___   Level 06

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $2BE_{16} = 702_{10}$ words of program.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___CREGEN - Regenerate Screen Display - CRT_____Level 06_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   CREGEN regenerates the display on the CRT screen.

2. DETAILED DESCRIPTION

   CREGEN calls ENDSR to save the caller's return address in the P10 area of
   the ATT. If the terminal is in idle mode, CCLRAL is called to clear the
   screen (SETBIT) and CDIS is called to set the TOPT data definition bits
   to idle. Exit to the caller is via EXDSR.

   If the terminal is in interrogate mode, CCLRFD is called to clear the inter-
   rogate line of the screen (CRG15). A check is made for Read mode if not in
   Read mode, TUTMOD is called to display the aid line on the screen if required
   and if a valid format is in use. The current screen position in data area is
   converted to a two-byte screen address by CONSCR and the cursor is displayed
   there by the display driver VL2CRT (CDATA). Then control is transferred to
   SETBIT with definition of interrogate.

   If the terminal is not in idle or interrogate mode, the page number of the
   current field is calculated by GETPL and the page is displayed by CPAGE. Then
   control goes to CRG15 where the current mode check is made.

INTERFACE

1. ENTRY METHOD

   Return jump

2. INPUT PARAMETERS OR CONDITIONS

   1 = current ATT address

3. EXIT METHOD

   Return to caller

4. OUTPUT PARAMETERS OR CONDITIONS

   CRT screen is regenerated.
   TOPT date definition bits are set.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME__ CREGEN - Regenerate Screen Display - CRT _____ Level 06 _____

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ZERO ($22)
        ONEBIT ($8D)
        ENDSR ($3E)
        EXDSR ($8F)

    b. ATT

        MODE (2)
        FLDN (14)
        TOPT (33)
        RQ (65)

    c. TOPT

        CURSRD (3)
        BLOKS (14)

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    CMISC1
    CLEARS
    GETPL
    CPAGE
    CENDS
    VL2CRT

4. DATA AREAS USED

    ATT
    TOPT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $34_{16} = 52_{10}$ words.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME____CREGEN - Regenerate Screen Display - CRT_____Level 06___

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Dormant reentrant.

AA5641

DOCUMENT CLASS __CYBERDATA   IMS__

PROGRAM NAME___DAM - Disk Allocation Module_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   DAM has two entry points, GITRT and RELTRT. GITRT allocates one track to
   the requester. RELTRT releases a track back to the allocation pool,
   provided it is not logged in the Bad Track Table (CDT - 64_{10}).

   The CYBERDATA definition of a track is 16 sectors of 96 words each. A
   cylinder is defined as 16 tracks.

2. DETAILED DESCRIPTION

   a. GITRT

       1) Call ENTSR to save return address in Exit stack.

       2) Call SETLU to set up disk parameters:

               Disk logical unit number.

               Disk logical unit in diagnostic messages.

               Base sector address of allocatable over on this disk unit.

               Length of Track Reservation Table (TRT) entry.

               Core address of TRT entry.

               Disk address of TRT entry.

               Values for 70%, 80%, and 90% of allocatable area on this unit.

       3) (CKCYL). Beginning with the first cylinder in the TRT entry,
          search for an available track.

       4) (FULDSK). If there are no tracks available on this disk unit, and
          EST/DSKST/f is not set, set EST/DSKST/f and write message:

               DISK u FULL

          on the output comment device.

          Note: f = 0,1,2, or 3 (disk full flag for unit 0,1,2, or 3)
                 u = disk logical unit number

       5) (FULEXT). Set (A) = 0 and call EXTSR to exit.

       6) (GOTI). If a track is available, mark it assigned in the TRT entry,
          and increment the count of assigned tracks.

       7) If (FLAG80) = 0, and the number of assigned tracks exceeds 80% of
          the allocatable area on the disk, set (FLAG80) = 1 and write the
          message:

               DISK u MORE THAN 80% UTILIZED

          on the output comment device.

8) (CALSEC). Calculate the sector address of the assigned track.

9) (WRTTRT). Call WRTSEC to write the modified TRT to disk.

10) Set (A) = sector address of assigned track and call EXTSR to exit.

b. RELTRT

1) Call ENTSR to save return address in Exit stack.

2) Call SETLU to set up disk parameters (see 1.b).

3) Check if track being released is logged as bad. If so, set release bit 5, and go to step 8.

4) Clear track assignment list in TRT entry. Decrement count of assigned tracks.

5) If EST/DSKST/f is set (disk full), and the number of assigned tracks exceeds 90% of the allocatable area on the disk, go to step f. If now under 90%, reset EST/DSKST/f, write the message:

   DISK u LESS THAN 90% UTILIZED

on the output comment device, and go to step f.

6) If EST/DSKST/f is not set, (FLAG90) = 1, and the number of assigned tracks is less than 70% of the allocatable area on the disk, set (FLAG80) = 0, and write the message:

   DISK u LESS THAN 70% UTILIZED

on the output comment device.

7) Call WRTSEC to write the modified TRT to disk and call EXTSR.

8) Write modified Bad Track Table to disk and exit via EXTSR.

INTERFACE   GITRT, RELTRT

1. ENTRY METHOD

   GITRT and RELTRT are core resident programs.

   Each is called by means of the RTJ instructions.

2. INPUT PARAMETERS OR CONDITIONS

   GITRT: (1) = ATT base address

   RELTRT: (A) = Sector address of track to be released

              (1) = ATT base address

3. EXIT MODULE

    Exit from both routines is to the location following the RTJ which called the routine.

4. OUTPUT PARAMETERS OR CONDITIONS

    GITRT: (A) = 0, no track is available for assignment

                (A) ≠ 0, sector address of assigned track

                Q register is not preserved.

    RELTRT: No output parameters

                Q and A are not preserved.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        EXTBL ($E9)
        LPMASK (2)
        NZERO ($12)
        ONEBIT ($23)
        SIXTEN ($27)
        ZROBIT ($33)
        TEN ($46)
        SST ($47)
        ENTSR ($91)
        EXTSR ($92)
        RTNSR ($98)
        WRTSEC ($A8)
        CLCDWA ($90)
        WRTDWA ($AA)

    b. EST

        TRTI (11)
        RTRIDA (17)
        LUNDI (25)
        DSKST (29)

    c. ATT

        LUN (35)
        ATEMP (73)

    d. EXTERNALS

        SIBSEC   Start of Cyberdata data storage

        ABSADD   Calculation of addresses

2. MACROS USED

   FWRITE

3. PROGRAM DEPENDENCIES

   ENTSR   Save return address in Exits stack
   EXTSR   Exit to return address in Exits stack
   RTNSR   Return address from Exits stack

4. DATA AREAS USED

   EST   Extended System Table
   ATT   Active Terminal Table
   TRT   Track Reservation Table
   BTT   Bad Track Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $150_{16} = 336_{10}$ words.

2. RELOCATION TYPE

   DAM is core resident program.

   It is loader relocatable.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS _____ CYBERDATA IMS

PROGRAM NAME _____ DBT - Delete Batch(es) _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    To process the DBT command, whose function is to delete batches from
    the disk.

    The format of the DBT command is as follows:
        DBT, JJJJJJ, xxx, yyy (,B)
    where:
        JJJJJJ - job name
        xxx   - initial batch number
        yyy   - last batch number

    If "B" is entered all specificial batches not active on a key station are
    erased, if "B" is not specificial, only batches previously transferred to
    tape by the WBT command are erased.

2. DETAILED DESCRIPTION

    The program makes itself movable and enables the option to be terminated
    by cancel key.

    The whole command is processed; FLAG indicates whether ",B" was entered
    (FLAG=0) or not (FLAG=1). In case of syntactic errors, return is made
    to SV with a suitable error message. Validity of the batch numbers is
    also checked (initial batch number should be less (equal) than 999);
    return is made to SV in case of error.

    GETLJD is evoked to get index of entry in LJD. In case of error (no job
    with that job name was found) return is made to sup. Before work begins
    the cancel bit is checked; in case cancel was entered return is made to
    sup mode and a message is printed. Next bit 13 of SST+1 (SLOCK) is checked
    to see if THW command was entered and locked system, in which case DBT
    should wait until this bit is cleared.

    SPILL is invoked to get index of that batch entry in ABT, address of first
    track in batch, mode and locigal unit.

    The mode is checked; if the batch is active (entry mode, verify mode or
    maintenance(and ",B" was entered, a suitable message is printed. If it
    is not active and ",B" was entered an attempt to delete is made. If ",B"
    was not entered, the mode is checked for written (6); in case it fits an
    attempt to delete is made, otherwise the next batch is taken.

    If the corresponding disk is not assigned to the system, a message is printed.
    If assigned, the entry in the ABT is deleted, and the tracks of the batch
    are released (by use of LINKTR).

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____DDC - Delete Document_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    DDC is a supervisor function, used to delete an entry from the Legal
    Document Directory (LDD). Calling sequence is DDC,nnn (CR) where nnn
    is the number of the document to be deleted.

2. DETAILED DESCRIPTION

    DDC first enters SETMOV to enable program moving and then collects the
    parameters following the DDC command via GETALP and GETBIN in SCANV.
    If DDC is not followed by a comma or the document number given is not a
    valid number, goes to REJSUP to indicate "Invalid command format" (ERR9).
    If the document number is greater than 255 goes to REJSUP to indicate
    "Invalid document number" (INVALD). If the parameters are accepted all
    the ATT's in the System are scanned to find if any terminal which is not
    IDLE or supervisor uses this document. If such a terminal is found goes
    to INVALD, else enters CLCDWA to calculate the disk address of the
    requested document entry in the LDD and writes zero via WRTDWA in PIØ
    over the first word of the entry. Exit is to SV.

INTERFACE

1. ENTRY METHOD

    DDC is a supervisor function loaded by ENTPRG.
    Only entry point is DDC.

2. INPUT PARAMETERS OR CONDITIONS

    I = Current ATT address.
    Operator enters:

                    DDC,aaa (CR) (aaa = document number)

3. EXIT METHOD

    a. Jump to REJSUP with appropriate error codes.

    b. Jump to SV.

4. OUTPUT PARAMETERS OR CONDITIONS

    I = Current ATT address.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____DDC - Delete Document_____Level 6_____

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK ($2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)

    b. SST

        ATTB (10)
        ATTLNG (19)
        STEMP (20)

    c. EST

        LDDDA (9)

    d. TRANSFER VECTOR

        CLCDWA ($90)
        GETBIN ($99)
        GETALF ($9B)
        WRTDWA ($AA)
        SV ($B1)
        REJSUP ($B2)

    e. ATT

        SLOCK (1)
        BMOD (15)
        DOCNO (17)
        DOC (46)

2. PROGRAM DEPENDENCIES

    EXTERNALS -

    SETMOV

3. TRANSFER VECTOR ROUTINES

    PIO   - CLCDWA: Calculate word address of disk resident information
                WRTDWA: Write to disk according to word addressing
    SCANN   - GETBIN: Collect string of ASCII digits and pack them into
                            hexadecimal number
                GETALP: Get one ASCII character from ASCII string
    SUPER   - SV: Conclude supervisory function request
                REJSUP: Generates error message to supervisor

DOCUMENT CLASS ___CYBERDATA IMS+___

PROGRAM NAME_____DDC - Delete Document_____Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $39_{16} = 57_{10}$

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

AA5641

PRINTED IN THE USA

DOCUMENT CLASS ____CYBERDATA_____

PROGRAM NAME_____DELETE_____Level 06_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    DELETE processes the manual DELETE function.

    DELETE is used with the INTLK key to delete the current record. In
    entry mode it may be used only after the first character of a record
    is keyed, but before the last character is keyed. In Verify mode, the
    key can be used only at the beginning of a field. In Read and Verify
    modes it must be followed by the COR key.

    The DLT key may not be used in Read mode of it could have been used under
    the circumstances immediately preceeding transfer to read mode. The DLT
    key is also used to cancel an INSERT.

2. DETAILED DESCRIPTION

    DELETE checks the current terminal mode. If it is not Entry or Insert,
    DELETE is accepted only at the beginning of the field, else ERROR is
    invoked with error code 4. While performing Replace field, DELETE is not
    accepted and ERROR (4) is called.

    In Verify or Read mode, DELETE checks if current and previous records are
    the same. If they are it calls ERROR with code 4. In Read mode, the
    number of records in Read should be more than one, or ERROR 4.

    Verify and Read mode will not allow a DELETE if there is only one record
    in batch, or if COR/DEL is already on.

    IDK and switch 1 (11),(1) are set for DEL/COR and COR must be next. Then
    CLRSR is called. CORKEY will make more checks and change pointers for
    deleted recored.

    In Entry mode DELETE uses the internal subroutine GETPFD to move back to
    the beginning of the record, and to reverse counters (using the external
    subroutine count). If autosequencing is specified, it is decremented by
    one, or set to gggg if it has already been zero. CMODE is set to IDK,
    and switch 1 (2) is turned on to indicate "first character in record".
    If the deleted record is not the first in batch, the previous record is
    brought to the TRB, and if there is an inserted record before the last
    one, the pointer to the next record is updated, and the ATT is written.
    The format is checked, and if it is different from the format of the deleted
    record, Switch 1 (15) is set to indicate "First record in format". Switch
    2 (4) is cleared to eliminate special processing by DUP. Next, record
    information is displayed, and DELETE exits via CLRSR.

If an inserted record is deleted, it moves to beginning of record, reversing counters. The bits for rewrite, insert, cor, skip key must be next, and first character in record in switch 1 are turned off. Switch 2 (15) (first nonzero) is cleared. Pseudo entry and pseudo verify flags in CMODE are cleared. If current disk address equals previous disk address in TRB, DELETE calls EXDFUN to free itself, and jumps to SETNFM in P10, for updating previous and next record pointers in TRB. If they are not the same, next record is read back to TRB, the corresponding format is brought in by GFCORE, and the date type of first field is fetched by STTYPE. After DISPLAY via DBSREL, DELETE exits to CLRSR.

INTERFACE

1. ENTRY METHOD

   DELETE is loaded via ENTPRG when the operator hits the DLT key.

   Execution begins at the only entry point DELKEY.

2. INPUT PARAMETERS OR CONDITIONS

   1 = Current ATT base address.

3. EXIT METHOD

   JMP-   (ERROR) with error code in Q, when an error occurs.

   JMP-   (CLRSR) when execution terminated normally.

   RTJ   (EXDFUN), JMP + SETNFM (in P10) - When insert.

4. OUTPUT PARAMETERS OR CONDITIONS

   I = Current ATT base address.

      Error message:

      $4 ILLEGAL FUNCTION KEY - (see detailed description, when this error
                                 is displayed).

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK   ($2)
      ONEBIT   ($23)

	ZROBIT	($33)
	ZERO	($22)
	NZERO	($12)
	ERROR	($96)
	DISPLY	($A0)
	CLPSR	($93)
	READP	($A5)
	GFCORE	($9D)
	REWRIT	($A4)
	READN	($A6)
	EXDFUN	($9F)
	SST	($47)
	WRTATT	($AC)
b.	SST	
	CFFA	(12)
c.	ATT	
	CMODE	(2)
	FCNT	(3)
	RCNT	(4)
	ATRB	(6)
	PFBA	(7)
	BMOD	(15)
	FMTNO	(13)
	RCD	(32)
	RCDER	(30)

DOCUMENT CLASS _____ CYBERDATA _____

PROGRAM NAME_____ DELETE _____ Level 06 _____

           RCDRD   (31)

           ATEMP   (73)

           SW1   (19)

           SW2   (20)

           AUSQ   (5)

           CURDA   (9)

           BMDA   (17)

           NDA   (11)

           BRCNT   (15)

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    EMM - COUNT - Subtract the contents of current field from the counter.

          STTYPE - Extracts data type from format and puts it into ATT.

    GFCORE - Releases the previous format and gets the one specified in Q register.

    FMTPAR - Extracts information from a format.

    DISPLAY-DEOREV - Displays end of record entry.

          DBSREL - Displays backspace record, Release.

    CLRSR - Clears the address stack and exits to dispatcher.

    P10 - SETNFN - Updates previous and next record.

          READP - Reads previous record.

          REWRIT - Reads next record.

          WRTATT - Writes ATT checkpoint.

    EXDFUN - Exits disk resident function.

    GSE - ERROR - Prints error message and exit.

4. INTERNAL SUBROUTINES

    GETPFD - Get previous information, and perform reverse counter operations
             as necessary.

5. DATA AREAS USED

    SST

    ATT

    TRB

    FBA

PHYSICAL CHARACTERISTICS

1. MEMORY SPARE REQUIREMENTS

    $(109)_{16} = (265)_{10}$ program length.

2. RELOCATION TYPE

    Run anywhere, not movable.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS _____CYBERDATA_____

PROGRAM NAME_____DER - Display Error_____ ⁴ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   DER is an interrogate function to display the error message associated with
   the code which appears on the screen when an error occurs.

   The command format is:

       INT DER, XX REL

   Where XX is a two character error code.

   Special processing is performed if SM is requested and there are supervisory
   messages to be displayed.

2. DETAILED DESCRIPTION

   SETMOV is invoked to allow the program to move, and ATT/SW1/8 is set to cause
   regeneration of the screen. The command format is checked via GETALP, and the
   two-character code is extracted. If the format is invalid, or non-ASCII
   characters were entered, exit is to ERROR (Illegal Parameter). The two
   characters are stored in the local variable LOCAL, and are compared with the
   table of valid codes (contained in ERRTAB). If no match is found, exit is to
   ERROR (Illegal Parameter). If a match is found but no SM, Q is converted to
   an index into the system table ERRM, and the entry is read from disk and
   displayed via DLINE. ATT/SW1/7 is set (RESET must be next), and exit is to
   CLRSR. If SM has been entered, but no messages have been sent, then processing
   is as detailed previously. If, however, there are messages to be displayed,
   a 96-word interrogate buffer is requested via GETBUF, and the message sector
   for the current terminal is read from disk into it. Next the message index
   is set in ATEMP+1 to allow display of up to eight messages in one pass. The
   screen is cleared, and the messages are displayed on successive lines until
   either all the messages have been shown, or the first eight. Then GETSTR is
   entered to await input of either RESET or CNCL. If either is entered, the
   display is regenerated, ATT/SW1/8 is cleared, and exit is to CLRSR. Otherwise,
   GETSTR is reentered to wait for one of these functions.

INTERFACE

1. ENTRY METHOD

   DER is entered via ENTPRG from INTRGT when the corresponding set of
   characters are keyed.

2. INPUT PARAMETERS

   I = current ATT base address.

3. OUTPUT PARAMETERS

   I = current ATT base address

   If no errors were detected, but code other than SM, or SM with no messages:

   - Interrogate line shows error message requested
   - ATT/SW1/7,8 are set.

   If SM was entered and messages were sent:

   - up to 8 lines of up to 24 characters each consisting of messages from the
     supervisor to the terminal.

4. EXIT METHOD

   a. Error exit is to ERROR with Q = $13 (Illegal Parameter)

   b. Normal exit is to CLRSR.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE
      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)
      SST ($47)

b. SST

STEMP (20)
CLCDWA ($90)
CLRSR ($93)
ERROR ($96)
GETALP ($9B)
DISP ($A0)
GETSTR ($A2)
RESSEC ($A7)
REDDWA ($A9)

c. ATT

SW1 (19)
AINT (27)
MSGP (78)

d. CONSTANTS

CE2 ($13)
RES ($67)

2. MACROS USED

None.

3. PROGRAM DEPENDENCIES

ENEXSR - SETMOV
         CLRSR
DISPLY - DLINE
         SETDIS
         CLRALL
         REGEN
PIO   - REDDWA
         REDSEC
         GETBUF
         GETSTR
         CLCDWA
GSE   - ERROR
SCANV - GETALP

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ DER - Display Error _____ Level 06

4. OTHER EXTERNALS

    ERRTL
    EPM
    SIERRM
    SIBMSG

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $AO_{16} = 160_{10}$ program length.

    96_{10} buffer for messages.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The display routines entry program routes all requested display functions
    via two parallel jump tables, one for the self-scan (TABMIT) and one for
    the CRT (TABCRT). There is only one program entry point - DISPLY. It is
    core resident.

2. DETAILED DESCRIPTION

    (DISPLY). If the function requested is DLOCK go to DISP1. Otherwise save
    A,Q in ATT/RQ and ATT/RQ + 1 and clear the 'page display' flag in ATT/IOADDR
    (word 0 of CRW).

    (DISP1). Set up the current return address and RTJ to ENTSR in order to save
    it in the ATT exits stack. Initialize the top of stack pointer to own
    stack in CRW (CRT work area in ATT).

    (DISP2). Check the display type of the device (ATT/SLOCK/10), get the
    address of the appropriate jump table, and RTJ to the requested function. On
    return, check the display type and if self-scan, go to DISP3. Otherwise,
    check the CRT for function lock on. If it is, exit via EXTSR. Next, check if
    the function requested was DCOL, DFMT, DDOC, DFNDTY, or DEOREV and if so, the
    cursor is repositioned via CONCSR and CDATA to its position before the function
    was executed. Then, clear the stack in P10 area in the ATT to indicate no I/O
    on terminal and return to caller via EXTSR.

    (DISP3). Self-scan exit: Inhibit interrupts and check if TOPT is threaded.
    If not, add it to the top of the Self-Scan thread in SST (word 5). Enable
    interrupts and return to caller via EXTSR.

    (DUMMY). Dummy display function: Where no subroutine exists for the required
    display function control is returned to caller via EXTSR.

INTERFACE

1. ENTRY METHOD

    RTJ through a transfer vector in SST (word $A0).
    I points to base address of current ATT.

2. INPUT PARAMETERS OR CONDITIONS

    In calling program, the next word after RTJ is the index of the desired
    display function. Upon entering, a RTJ to ENTSR is made. See Display
    Functions (Attach Analytical Supplements).

3. EXIT METHOD

    Exit is via EXTSR or a direct jump to address obtained from RTNSR

4. OUTPUT PARAMETERS OR CONDITIONS

    N/A

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        ZERO ($22)
        ONEBIT ($23)
        SST ($47)
        CDATA ($3D)
        ENTSR ($91)
        EXTSR ($92)
        RTNSR ($98)

    b. SST

        TOPCRT (4)
        TOPMT (5)
        TEMP (20)

    c. ATT

        T (2)
        DSP (10)
        TOPT (33)
        IOADDR (64)
        RQ (65)

    d. TOPT

        CURSRD (3)

    e. CRW

        C (1)
        THREAD (2)

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___DISPLY - Display Routines Entry Program_____Level 06___

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    VL2CRT (CONCSR)   Convert CRT address from a binary number to two 8-bit
                         codes in required format.

4. DATA AREAS USED

    The display routines use the P10 area in the ATT as a work area.
    TEMP area in SST.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $99_{16} = 153_{10}$ words.

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Dormant reentrant.

ATTACH ANALYTICAL SUPPLEMENTS

Attachment: List of display functions

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME__DISPLY - Display Routines Entry Program_____Level 06____

Display Functions:

The display function table is set up as a table of entry points which are
equivalenced to an index for the dual tables of self-scan and CRT. Thus:

Index	Function	Function Description
0	DCOL	Display Column Number
1	DERR	Display Error Code
2	DFMT	Display Format
3	DMOD	Display Mode
4	DBAX	Display Batch Number
5	DDOC	Display Document Number
6	DJOB	Display Job Name
7	DRCNT	Display Record Number
8	GENHDR	Generate Header Line Display
9	DBOF	Display Beginning-Of-Field
10	DCHAR	Display Character
11	DCAROL	Display Character and Column Number
12	DEOF	Display End-Of-Field
13	DEOREV	Display End-Of-Record Entry/Verify
14	DBORKEY	Display BOR Key
15	DBSCEV	Display BSC Key Entry/Verify
16	DBSCR	Display BSC Key Read Mode
17	DBSF	Display BSF Key
18	DSKPR	Display SKP Key Read Mode
19	DBSREL	Display BSR/REL Keys Read Mode
20	DREADK	Display Read Key
21	DSKUP	Display Manual SKP/DUP
22	DSPACE	Display Space Key Read Mode
23	CLRALL	Clear Total Display
24	CLRFLD	Clear Field
25	REGEN	Regenerate Display
26	DLINE	Display Line
27	SETDIS	Set Display Definition Bits
28	DLOCK	Display System Lock
29	DMLDS	Display Multi-Level SKP/DUP
30	DFNDTY	Display Field Number and Data Type
31	DSUP	Display -SV- On CRT
32	DSPFLD	Display Field For Sight Verification
33	DSPDIR	Display Direct 5-Word Buffer (Address of buffer in RQ,I)
34	DMSG	

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____DJB - Delete Job_____Level 6_

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

DJB is entered by the supervisor when a job has to be deleted from the system.

2. DETAILED DESCRIPTION

a. DJB starts by making itself movable in the memory and checking if the DJB command is followed by a comma. If not, an error message is sent to the supervisor and the command is rejected.

Next, a temporary area (JOBNAM to JOBNAM+2 in ATT) is blanked and a counter (ATEMP) is set to zero. ATEMP counts the number of input characters in the job's name.

A loop (headed NXTCHR) reads in one character at a time and stores it in its corresponding place (in JOBNAM) using bit 0 of ATEMP as an indicator in which half to store the character. If EOR is encountered, go to b). Any illegal ASCII character in the input buffer results in an error message and rejection of the command. After 6 characters were read in, a test is made (through the GETCOM subroutine) to find the End-Of-Record indicator. Not finding it (existence of more characters) results in an error sent and rejection of the command.

b. After the job name is filled into ATT/JOBNAM, it is passed as a parameter to the GETLJD subroutine, with Q set to 1 to obtain the corresponding LJD entry. If the Job does not exist (Q = 0 at exit from GETLJD) the command is rejected and an error message is sent.

If found, the Job Index (returned in A) is stored in the relevant word of the ATT (word 3).

The job index is passed to the SPILL subroutine together with an indication for checking the job activity. If SPILL returns a nonzero value in A (ABT for that job still active), an error message is printed and the command is rejected. If not, a call is made to the GETLJD subroutine, to delete the LJD entry for that job, and control is returned to SV.

INTERFACE

1. ENTRY METHOD

From DSKSUP, through ENTPRG.

2. INPUT PARAMETERS

I = current ATT base address.

3. EXIT METHOD

If any error detected, exit through REJSUP.
If no errors, return to SV.

4. OUTPUT PARAMETERS

    I = current ATT base address (not changed).
    Corresponding entry in LJD detected (if no errors detected).

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        SST ($47)
        GETCOM ($9C)
        SV ($B1)
        LPMASK ($2)
        GETALP ($9B)
        REJSUP ($B2)
        SPILL $A1)

    b. ATT

        STEMP (20)
        JIDX (3)
        ATEMP (73)
        SUPCOM (46)

    c. CONSTANTS

        None

    d. EXTERNALS

        GETLJD,SETMOV

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    GETALP
    SV
    GETLJD
    REJSUP
    SPILL
    GETCOM

4. DATA AREAS USED

    SST
    LJD
    ATT
    ABT

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __DJB - Delete Job__ _____ __Level 6__

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $53_{16} = 83_{10}$ words.

2. RELOCATION TYPE

    Loader relocatable, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME____DMP1_____Level 06____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The program is a supervisor function which analyzes the DMP commands and
    saves the parameters for the next overlay in the communication area in the
    ATT.

    It searches for the current place on the tape for writing. Two different
    overlays can be loaded and executed in case of different DMP parameters.

2. DETAILED DESCRIPTION

    The program gets as valid input in the interrogate buffer the following
    strings:

    DMP,J $\left[,jjjjjj\right]$ $\left[,m\right]$

    DMP,D,n $\left[,m\right]$

    DMP,S $\left[,n\right]$

    DMP,F.........

where: J stands for jobs

         D stands for disk (all jobs of disk n)

         S stands for statistics file

         F stands for formats

    jjjjjj - job name. Default value is all jobs.

         m - mode of batches. Default value is all batches.

              F - frozen batches

              C - completed batches

              W - written batches

         n - disk number

         r - delete/retain parameters. D - delete; R - retain.

             Default value is R.

           - means the parameter is optional

Note: The parameters of DMP,F are checked by the second overlay DMPFMT.

The first part of the program which performs the syntax analysis is locked.
Depending upon the value of the first parameter, a branch is taken to the
section which performs the specific checks for the parameters to come. If
the second parameter is J it is verified that if a job name is specified,
it corresponds to a legal job in the system. If the second parameter is D,
it is verified that the disk number corresponds to a selected disk. In both
cases it is verified that the mode of batches is one of the legal possibilities.
If the second parameter is S, the retain/delete parameter is checked. If the
second parameter is F, the second overlay (DMPFMT) is loaded and executed.
If any of the checks fails, the program exits through the common error exit
of the first part of the program. The possible errors are "invalid parameter"
for any syntax error, "invalid job name" if the job name which is specified
in the command does not exist in the system, "disk n is deselected" if this
is the situation. In the common error exit the lock of the first part is
cleared and the program exits to REJSUP.

If all the checks pass successfully, the following variables in the mag tape
communication area are assigned the following values:

    DT - word 1

    0 - dump all jobs in system

    1 - dump specific job

    -1- dump disk

    $FFFF - dump statistics

    NO - word 2

        disk number if DMP,D

        job index if DMP,J, job name

    BMODE - word 3

    if DMP,S:

                ASCII value of D   if DMP,S,D

                ASCII value of R   if DMP,S,R

                0 - no retain/delete parameter was specified

if DMP,J or DMP,D:

			W	C	F

bit 0 - 1-Dump frozen batches

bit 1 - 1-Dump completed batches

bit 2 - 1-Dump written batches

0 - Dump all batches

In the second part, the input tape is checked to have valid structure for DMP operation, and the free section for writing is searched. A valid tape for DMP operation can be a new tape with the tape label EOD1 preceded by a tape mark or a tape which was written previously by a DMP operation.

The following structures are valid:

TM means tape mark. The alphanumeric characters mean their ASCII values. The tapes should be written in binary. All the labels are 16 words long. All other tapes are rejected.

The checks are made as follows: If the tape is positioned by previous DMP operation rewind of the tape should be avoided. The tape is back-spaced two records and a check is made to see if both were tape marks. If so, a file is backspaced and a record advanced to position the tape before the label of the last DMP operation. If the label is a valid dump header, the file number in the mag tape communication area is updated with the cor-responding value in the header. If it is not a DMP header which matches the data type which was specified in the command, the tape is rewound and a check is made to see if it has in the beginning of a system label EOD1 by a tape mark, or appropriate dump header. The tape is rejected if neither case per-tains. If the tape is output of a previous dump operation, the tape is

positioned between the two ending tape marks. The file number in the mag
tape communication area is updated to the number of the files skipped. The
terminate bit is checked every skipped file.

If it is a new tape (with system label EOD1), it is positioned at the begin-
ning of the tape.

The tape handling section has 4 error exits to REJSUP.

After each tape motion request, the status is checked. If "end of tape" or
"tape I/O error" were encountered during the motion request, exit is taken to
REJSUP with an appropriate error code. If tape I/O error was encountered while
reading the headers or if the tape did not have a valid structure, exit is to
REJSUP with "wrong tape" message error code. If the terminate bit was set
while reaching the tape, exit is to REJSUP with "function terminated" message
error code.

In the normal exit, the program releases itself and loads the next overlay
DMP2.

INTERFACE

1. ENTRY METHOD

    The DMP1 program is loaded and executed via ENTPRG by the disk resident
    portion of SUPER, DSKSUP.

2. INPUT PARAMETER OR CONDITIONS

    DMP command string in the interrogate buffer.

3. EXIT METHOD

    a. Normal Exit.

        1) If the command string begins with DMP,F the DMPFMT overlay is loaded
            and executed via ENTPRG.

            CLR   A

            LDQ   =N$0810   load and execute 16'th miscellaneous program

            RTJ-   (ENTPRG)

        2) In all other cases if the syntax analysis and the tape analysis pass
            successfully, the DMP2 overlay is loaded and executed via ENTPRG.

b. Error Exit.

Error exit is via REJSUP with appropriate error message code in A register.

JMP- (REJSUP)

4. OUTPUT PARAMETERS OR CONDITIONS

Normal exits:

a. The parameters for the DMPFMT overlay in the command string in the interrogate buffer.

b. Parameters for DMP2 overlay:

Mag Tape Communication Area (MTCA) in ATT DT - word 1 - dump type.

0 - dump all jobs

1 - dump specific job

-1- dump disk

$FFFF - dump statistics

No - word 2

    disk number if DT = -1

    job index   if DT = 1

BMODE - word 3

    if DT = $FFFF

    ASCII value of R or D

    Binary zero

    if DT = 0,1,-1

    bit 0 = 1 - dump frozen batches

    bit 1 = 1 - dump completed batches

    bit 2 = 1 - dump written batches

    Binary zero - dump all batches

FILEN - word 4 - file number

Number of tape marks until the point at which the tape is positioned.

Error exit:

Possible error codes:

3 -   invalid parameter

0 -   invalid job name

65+n - disk n is deselected

58 -   tape I/O error

59 -   end of tape

70 -   wrong tape

39 -   function terminated

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK   (2)

        NZERO   ($12)

        ZERO   ($22)

        ONEBIT   ($23)

        ZROBIT   ($33)

    b. SST

        SST   ($47)

        TEMP1   (20)

        TEMP2   (21)

    c. EST

        DSKST   (29)

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME____DMP1_____Level 06___

    d. ATT

        TRMNT   (23)

        IRB   (27)

        MTCA   (46) - mag tape communication area

        ATEMP   (73)

        DT   (MTCA) - dump type

        NO   (MTCA+1) - disk number or job index

        BMODE   (MTCA+2) - batch mode or retain/delete parameter

        FILEN   (MTCA+3) - file number

    e. TRANSFER VECTOR

        ENTSR   ($91)

        EXTSR   ($92)

        ENTPRG   ($9E)

        EXDFUN   ($9F)

        SUPRW   ($97)

        GETBIN   ($99)

        GETALP   ($9B)

        TAPIO   ($9A)

        GETCOM   ($9C)

        REJSUP   ($B2)

        WAIT   ($B0)

    f. GENERAL EQUATES

        FR (2)

        FW (3)

        MO (4) - tape motion request

BR (8) - binary read

J ($4A)⎫

D ($44)⎪

S ($53)⎪

R ($52)⎪

A ($41)⎬ ASCII values

F ($46)⎪

C ($43)⎪

W ($57)⎭

2. MACROS USED

None.

3. PROGRAM DEPENDENCIES AND ROUTINES USED VIA TRANSFER VECTOR

	ENTSR	
	EXTSR	
ENEXSR	SETMOV	- enable program's moving
	WAIT	- used if syntax analysis section part is locked
GETLJD	used to get the job index of the job specified in the command
PIO	TAPIO	- used for motion and read requests
	SUPRW	
	ENTPRG	- used to load next overlays
ENTPRG	EXDFUN	- releases the program
	GETALP	- used for syntax analysis
SCANV	GETCOM	- used for syntax analysis
	GETBIN	- used for syntax analysis
SUPER	REJSUP	- used in case of error exit

AA5641   PRINTED IN THE

4. DATA AREAS AND TABLES USED

SST

ATT - MTCA (Mag Tape Communication Area)

EOD1

DMP header - see Appendix A of DMP2

IRB - interrogate buffer

PHYSICAL CHARACTERISTICS

1. PROGRAM LENGTH

    154_{16} words.

2. RESIDENCE TYPE

    Disk resident.

3. RELOCATION TYPE

    Run anywhere, movable.

4. REEANTRANCY

    Reentrant, partially locked.

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The program dumps the statistics file or batches as requested in the DMP
    command. The output tape is positioned on the point from which writing
    should begin. Informatory messages are printed.

2. DETAILED DESCRIPTION

    The program is the second overlay of DMP1. It is responsible for handling
    the dump operation of the statistics file and batches. The dump type and
    other parameters are transferred to the program in the magnetic tape
    communication area.

    The program is constructed from two separate sections. One deals with the
    statistics file and the other with batches. The section dealing with the
    statistics file is locked. For both purposes a buffer is allocated for the
    dump operation. The statistics file dumping needs a core buffer of 1 sector
    (96 words) length. The batches dumping needs a buffer 4 sectors long (an
    equate in the program) plus 91 words for the TKT (Tracks Table). The buffer
    is connected to the TRB backwards pointer. If there is no core available
    for the buffer, the program exits to REJSUP with "memory full" message error
    code. A message informing the supervisor from whose station the file dump
    is being performed, is printed.

    The statistics file dumping proceeds as follows:

    If number of records in the statistic file (in EST) equals zero, an informatory
    message is printed and the program exits through the error exit. If the file
    is not empty, DMP header is written on the tape (see Appendix A).

    The statistics file is read sector after sector (via internal subroutine READ),
    to the core buffer and every sector is searched to see if end of statistic
    file mark (binary zero in the first word of a statistic record) is present.
    If not, the whole sector (6 statistics records) is written to the tape.
    Otherwise, the empty records are padded with $FFFF, and the whole sector
    is written to the tape. If end mark was encountered or if the last sector
    of the statistic file was written (its address is found in the EST), 2 tape
    marks are written to the tape and the tape is rewound and unloaded.

    If the delete option was specified or if the last sector was full, the request
    priority is increased by 1 and an end of statistics file (binary zero) is
    written in the beginning of the statistics file. If the number of records
    specified in the EST exceeds the number of records until the end of the mark,
    the tape is positioned 3 records back (2 tape marks and 1 sector), the last
    sector is read again to the core buffer and the process continues as before.

The statistics file handling section has a common error exit with the batch
handling section. Possible error messages are "end of tape" and "tape I/O
error". Errors are detected by CHECK1 subroutine, called after every tape
request. Before writing to the tape is performed, the terminate bit is
checked (in the internal subroutine TERMIN), and "function terminated" is
printed if the bit is set. The messages are printed via PRTMSG. In the
common error exit the tape is positioned before the actual file. The lock
is cleared and the TRB is released (via internal subroutine RELEASE), and
the program exits via REJSUP. Normal exit is to SV after clearing the
lock and releasing the TRB (via RELEASE).

The batches dumping proceeds as follows:

If all the batches of all jobs or all the batches of a certain disk should
be dumped, the LJD is searched from beginning to end. The job entry is
looked for through GETLJD with job indices running from 1 to 255. If an
index is missing, the procedure continues with its successor. If only the
batches of a certain job should be dumped, this job is looked for through
GETLJD. Its job index is transferred to this overlay via the mag tape
communication area. The batches of each job are checked. The ABT entries
of the batches with numbers running from 1 to 999 are brought through SPILL.
If a number is missing the procedure continues with the closest existing
successor (returned by SPILL).

For every batch a check is done to find a match to the mode of batches
specified in the command. If no mode was specified all inactive batches are
dumped. If the command was DMP,D (DT = -1) a check is made to see if the
disk number on which the batch resides is equal to the disk number which was
specified in the command. If the command was DMP,J (DT = 0,1) a check is
made if the disk on which the batch was written is selected. If the disk
is deselected a message informing the supervisor that the batch is rejected
is printed and the procedure continues with the next batch. If the batch is
in maintenance mode an informatory message is printed, and the procedure
continues with the next batch. Before a batch is dumped it is assigned to
maintenance mode. Every batch is dumped as a file. An informatory message
with the job name and the batch number is printed.

The file number in MTCA is increased by 1 and the DMP header is written
to tape. The tracks table is read (via READ routine) to the beginning of
core buffer. The table is scanned. Every track is read (via READ) 4 sectors
(an equate in the program) at a time to the core buffer after the tracks
table, and the 4 sectors are written to the tape. When a zero entry in the
tracks table is encountered, 2 tape marks are written to the tape and the

tape is positioned after the first one. The batch's mode is changed from the maintenance mode to its original mode. If another batch is dumped, it overwrites the second tape mark. If all the batches specified in the command were dumped, the tape is advanced one record and is positioned after the second tape mark. The TRB is released (via RELEASE subroutine) and the program exits to SV.

Errors which abort the dump operation are errors connected with tape I/O. After every I/O operation the tape is checked (via CHECK2 routine). If end of tape was encountered the batch is assigned at its original mode.

The tape is positioned before the last tape mark, (the last file is backspaced). An end of volume label (see Appendix A) is written to the tape. The message "end of tape - mount new tape and type CU" is printed, and the program reads to the interrogate buffer the answer of the supervisor. If it is no "CU" the program loops, prints the message and waits for the appropriate answer. If the answer is "CU" the new tape is rewound and the procedure continues. If an I/O error was encountered, the batch is assigned its original mode. Exit is taken through the I/O error exit in the statistics file handling section. Every batch, the terminate bit is checked (via TERMIN subroutine) and exit is done through the common error exit.

All the messages in the program which are not printed through PRTMSG, are printed from the interrogate buffer in the internal subroutine WRITE.

INTERFACE

1. ENTRY METHOD

   The program is loaded and executed by DMP1, which is initiated by the supervisor command DMP.

        LDQ   =N0809

        CLR   A

        RTJ-   (ENTPRG)

2. INPUT PARAMETERS OR CONDITIONS

   Several parameters are transferred by DMP1 in the MTCA (Mag Tape Communication Area).

Word 1 - DT - dump type

$FFFF - dump statistics file

0 - dump all batches in the system

-1 - dump disk

1 - dump job

Word 2 - NO -

if DT = -1 , disk number

if DT = 1 , job index

Word 3 - BMODE

if DT = $FFFF , 0 , ASCII D or R

if DT = 0,-1,1   , batches mode

  bit 0 = 1   , dump frozen batches

  bit 1 = 1   , dump completed batches

  bit 2 = 1   , dump written batches

Word 3 - FILEN

number of tape marks up to the point the tape is positioned at

3. EXIT METHOD

a. Normal Exit

The normal exit is through SV.

JMP- (SV)

b. Error Exit

Error exit is through REJSUP

JMP- (REJSUP)

DOCUMENT CLASS CYBERDATA

PROGRAM NAME DMP2   Level 06

4. OUTPUT PARAMETERS OR CONDITIONS

   The dumps operation can be aborted due to several conditions:

   If dump statistics:

                        1. Statistics file is empty

                        2. End of tape

   For all dump operations:

                        3. Function was terminated

                        4. Tape I/O error

                        5. No memory available for a buffer

   The following messages are printed corresponding to the errors:

                        1. STATISTICS FILE IS EMPTY

                        2. END OF TAPE

                        3. FUNCTION TERMINATED

                        4. TAPE I/O ERROR

                        5. MEMORY FULL - ENTRY

   Messages 2, 3, 4 are printed via the external routine PRTMSG. The messages
   correspond to the codes 59, 39, 58.

   After these errors occur, the tape is positioned before the last erroneous
   file and 2 tape marks are written to the tape.

   Message 5 is printed by REJSUP. The error code is 11. Informatory messages
   are printed during the dump operation.

   When the operation is started:

                        DUMP STARTED, F!LE nnnn

                   where: nnnn is the number of the file.

For every batch whose status and disk number matches the status and the disk number specified in the command, an informatory message is printed.

If the batch is dumped:

>   bbb jjjjjj

If the batch is rejected:

>   bbb jjjjjj REJECTED; BATCH IS IN MAINTENANCE MODE
>
>   bbb jjjjjj REJECTED; DISK n IS DESELECTED

where: bbb   -   batch number

　　　　jjjjjj -   job name

　　　　　　n   -   disk number

If end of tape is encountered while dumping batches, the following message is printed:

>   END OF TAPE - MOUNT NEW TAPE AND TYPE CU

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK   (2)

        NZERO   ($12)

        ZERO   ($22)

        ONEBIT   ($23)

        ZROBIT   ($33)

b. SST

    SST   ($47)

    TEMP1 (20)

    TEMP2 (21)

    RPF   (18) - request priority

c. EST

    SFDA   (5)

    ESSF   (6)

    RCSSF (7)

    DSKST (29)

d. ATT

    FTOB   (1)

    JIDX   (3)

    TRB   (6)

    ABTIDX (22)

    TRMNT (23)

    IRB   (27)

    BATCH (35)

    MTCA   (46)

    ATEMP (73)

Communication Area Equates:

DT   (MTCA)   ‒   dump type

NO   (MTCA+1)   ‒   disk no. or job index

BMODE (MTCA+2)   ‒   batch mode

FILEN (MTCA+3)   ‒   file number

COUNTR (MTCA+4)   ‒   statistics file records counter

NXTDA (MTCA+5)   ‒   next disk address

NOSEC (MTCA+6)   ‒   sector counter

NOTRK (MTCA+7)   ‒   track counter

BN   (MTCA+8)   ‒   batch number

JN   (MTCA+9)   ‒   job name

DTEMP (MTCA+12)   ‒   temporary

e.   GENERAL EQUATES

BLOCK (4)   batch buffer length in sectors

FR   (2)   SUPRW fread

FW   (3)   SUPRW fwrite

MO   (4)   motion request code of the tape

BW   (9)   binary write

D   ($44)   ASCII value

f.   TRANSFER VECTOR

ENTSR ($31)

EXTSR ($32)

RTNSR ($38)

SUPRW ($97)

TAPIO   ($3A)

EXDFUN ($9F)

SPIL   ($A1)

REDSEC ($A7)

WRTSEC ($A8)

RELFOR ($AD)

CAM   ($CAM)

WAIT   ($B0)

SV   ($B1)

REJSUP ($B2)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES AND ROUTINES USED VIA TRANSFER VECTOR

ENEXSR
- ENTSR
- EXTSR
- RTNSR
- SETMOV
- WAIT - used if dump statistics section is locked, or if core is not available in this cycle.

PIO
- MOVREC
- SUPRW - used for printing the messages, reading the answer
- TAPIO - used for tape handling
- REDSEC - read statistics file and the batches from the disk
- WRTSEC - write first sector of the statistics file with end mark (binary zero) in the beginning.

PAGE NO. _153__

CAM   -   allocate core buffer for dump operation

SUPER   { REJSUP

          SV

          PRTMSG -   print messages from reject messages area

GETLJD

ENTPRG { EXDFUN

          ENTPRG

SPILL -   used for all ABT processing

GFCORE { RELFOR -   used for releasing the TRB

ARITH   { DB   -   used for batch numbers manipulation

          BD   -   used for batch numbers manipulation

System externals

ADAYTO   -   day of month in ASCII

AMONTO   -   month in ASCII

4. DATA AREAS AND TABLES USED

SST

ATT

SRL   -   statistics file

ABT

TKT   -   tracks table

IRB

DMP header

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___DMP2___   Level 06

PHYSICAL CHARACTERISTICS

1. PROGRAM LENGTH

    $30C_{16}$ words.

    Buffer length - $91_{10} = 96_{10}4_{10}$ words.

2. RESIDENCE TYPE

    Disk resident.

3. RELOCATION TYPE

    Run anywhere, movable.

4. REENTRANCY CLASSIFICATION

    Reentrant, partially locked.

AA5641   PRINTED IN THE USA

APPENDIX A

1. DMP headers

    a. Statistics

        DMPSMMDDFN00000000000

    b. Batches

        DMPBMMDDFNJJJJJJNBNFTOOOOO

        where: DM\ - ASCII value of the two letters

                    PS - ASCII value of the two letters

                    PB - ASCII value of the two letters

                    MM - ASCII value of the month

                    DD - ASCII value of the day of month

                    FN - binary value of the file number

            JJJJJJ - job name (in ASCII)

                    JN - job index (in binary)

                    BN - batch number (in packed decimal)

                    FT - first track of batch address.

2. End of volume label

    EOV1............................

    where EO - ASCII value of the two letters

            V1 -

3. DMP operation output tape

    a. Statistic

        DMPSMMDD0001   ........S.F.......TM DMPSMMDD0002·   ....S.F.   TM....TM TM

        where S.F. is the statistics file information.

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___DMP2___   Level 06

    b. Batches

DMPBMMDD0001JJJJJJ000100012000...... ....batch # 1...TM DMPBMMDD0002JJJJJJ00010002

.....in batch # 2.....TM .... TM TM

DOCUMENT CLASS _____ DMPFMT _____

PROGRAM NAME _____ Dump Format From Disk To Tape _____

PROGRAM FUNCTION

1. ENTRY POINTS

   There is only one entry point to the program (ADC - DMP-DMPHDR)

2. DETAILED DESCRIPTION

   Functional Description

   The entered command and format numbers are checked not to contain any input
   errors. Tape is checked to be the right one. Either tape's header or tape
   mark is searched. If the latter is found, the conclusion is - new tape. If
   any header is found it is then checked to be the proper one: header containing
   date and last block size used. If not found, then a WRONG TAPE message is sent
   to supervisor. The same message is sent when EOD1, which indicates the end of
   last written data, is not found. Request is made to CAM for three buffers:
   2 sectors for LFD, 34 words for building the directory to tape and up to 16
   sectors for formats save - in case it is not a new tape, the formats save
   buffer length is saved according to last printed on tape.

   Format is checked in the LFD. If the format exists it is transferred from
   disk to formats buffer according to format length in words. If not specified,
   the next in the LFD is checked, unless otherwise requested. When one of the
   buffers, either the formats buffer or the directory buffer is full, it is
   transferred to tape.

   In case of a new tape, suitable header is written at the beginning of the
   tape. If the last format requested is already reached, buffers are transferred
   to tape although they might not be full, and EOD1 is placed at the end of the
   buffer.

   A message is printed indicating operation is completed and the program exits
   to -SV-.

INTERFACE

1. ENTRY METHOD

   The supervisor enters the following command:

$$
\text{DMP,F} \left\{ \begin{array}{l} \text{,nnnn(1)} \\ \text{,nnnn(1),nnnn(2)} \\ \text{all} \end{array} \right\} \text{CR}
$$

   where: F   - Formats dump.
            nnnn(1) - first format number to be dumped.
            nnnn(2) - last format number to be dumped.
                      When last format number is not specified all the formats
                      from nnnn(1) to the highest format number in the system
                      will be recorded on tape.

355

DOCUMENT CLASS ___DMPFMT___

PROGRAM NAME_____Dump Format From Disk To Tape_____

When no format number is specified - all formats in system will be recorded on tape.

2. INPUT CONDITIONS

Requested format numbers, nnnn(1) and nnnn(2) must be positive integers (not zero). nnnn(2) can be greater or equal to nnnn(1) last format number can be greater or equal to the first format number. Neither nnnn(1) nor nnnn(2) can be greater than the highest format number in the system.

3. EXIT METHODS

The program can be terminated by one of the following methods:

Error exit - if an error is detected in the input parameters, in the logic structure of the request, or if tape is not identified as the requested one, program exits to REJSUP with error message.

Return to caller - when no errors are detected and operation is completed, program exits to -SV- with a completion message.

4. OUTPUT PARAMETERS

PROGRAM'S ERROR MESSAGES

a. INVALID COMMAND FORMAT - when comma does not follow commands name, or C R is missing at the end of the command.

b. INVALID FORMAT NUMBER - when requested format numbers do not fit the description in section 2 of INTERFACE.

c. TAPE I/O ERROR - when something is damaged in the tape operations (read, write, advance or rewind).

d. WRONG TAPE - when tape is not identified as the requested one.

e. END OF TAPE.

f. MEMORY FULL - RETRY - when CAM fails to get the requested word-size buffers.

g. FUNCTION TERMINATED - if MI and SX are pressed during program run time, function is terminated.

COMPLETION MESSAGES

1. LAST FORMAT ON TAPE NNNN: when NNNN is format number.
2. NO ACTIVE FORMATS WITHIN THIS RANGE IN SYSTEM.

3-56

DOCUMENT CLASS ___DMPFMT___

PROGRAM NAME___Dump Format From Disk To Tape___

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. Low Core Equates

      LPMASK ($2)
      NZERO ($12)
      ONEBIT ($23)
      ZROBIT ($33)
      ZERO ($22)
      TEN ($46)

   b. SST Equates

      SST ($47)
      STEMP ($5B)
      EXTSST (0)
      MAXFMT (24)
      LFDDA (39)
      GETBIN ($99)
      GETALP ($9B0)
      REJSUP ($B2)
      SV ($B1)
      CAM ($AE)
      RELFOR ($AD)
      WAIT ($B0)
      MAXWT (100)
      SUPRW ($97)
      CLCDWA ($90)
      WRIDWA ($AA)
      REDDWA ($A9)
      REDSEC ($A7)
      TAPIO ($9A)
      ENTSR ($91)
      EXTSR ($92)

   c. ATT Equates

      IRB (27)
      CANC (23)

3.5-7

DOCUMENT CLASS ___DMPFMT___

PROGRAM NAME_____ Dump Format From Disk To Take _____

   d. Communication Area Equates

    SUPCOM (36), SC (SUPCOM + 10)
    FMT1 (SC + 0)   - first requested format no.
    FMT2 (SC + 1)   - last requested format no.
    FMTN (SC + 2)   - last format no. on tape.
    LSTBLK (SC + 4) - last block size used in previous dump. (in words
                       exc. HDR).
    BUFSIZ (SC + 5) - big buffer size (inc. HDR).
    BUFF (SC + 6)   - backwards pointer of the LFDBUF.
    RELLFD (SC + 7) - relative address of the LFDBUF.
    BUFD1 (SC + 8)   - backwards pointer of the Directory Buffer.
    RELDIR (SC + 9) - relative addressing of the Directory Buffer.
    FMTWRD (SC + 10) - format length in words.
    BCKP (SC + 11)   - backwards pointer to big buffer.
    RELBIG (SC + 12) - relative addressing in big buffer.
    SAVE (SC + 13)   - flag for errors, contains error code.
    SUBF (SC + 16)   - any active format.
    FLAG (SC + 17)   - sign for reading tape, and for a new tape.

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   Entry points

   None.

   External list

  AYERTO   - Entry in SYSDAT.
                 Retrieve year in ASCII.
  AMONTO   - Entry in SYSDAT.
                 Retrieve month in ASCII.
  ADAYTO   - Entry in SYSDAT.
                 Retrieve day in ASCII.
  SETMOV   - Entry in ENEXSR.
                 Enables moving of a disk resident program.

   Routines called via transfer vector

  GETBIN ($99) - Entry in SCANV.
                 Retrieve the formats numbers.
  GETALP ($9B) - Entry in SCANV.
                 Retrieve the comma following command.
  REJSUP ($B2) - Entry in SUPER.
                 Exit when input command error occurs or when tape fails.

358

DOCUMENT CLASS ___DMPFMT_____

PROGRAM NAME_____Dump Format From Disk To Tape_____

```
    SV ($BI)      - Entry in SUPER.
                    Exit when command is accomplished.
    CAM ($AE)     - Entry in CAM.
                    Request core space for programs buffers.
    RELFOR ($AD)  - Entry in GFCORE.
                    Release the requested format.
    TAPIO ($9A)   - Entry in PIO.
                    Performs tape motion, read and write (from and to) specified
                    buffer.
    CLCDWA ($90)  - Entry in PIO.
                    Calculates disk word address.
    REDDWA ($A9)  - Entry in PIO.
                    Reads word addressing.
    REDSEC ($A7)  - Entry in PIO.
                    Reads sector addressing.
    SUPRW ($97)   - Entry in PIO.
                    Called to read or write from or to suitable logic units.
    ENTSR ($91)   - Entry in ENEXSR.
                    Saves the enter address of subroutines with I/O or delay in the
                    exits stack A, Q, and I are saved.
    EXTSR ($92)   - Entry in ENEXSR.
                    Exit subroutine used by subroutine which entered by ENTSR,
                    exit to last address in the stack.
    WAIT ($B0)    - Entry in ENEXSR.
                    Used to delay function for one cycle.
```

4. DATA AREAS AND SYSTEM TABLES USED

   LFD is read into a suitable buffer in order to check which formats exist
   and where.

5. INTERNAL SUBROUTINES

   CHKTAP - checks tape after any tapes operation: read, write, advance or
            rewind.

            Input parameters: after coming from tapio.
                               Register A - contains tapes status.

   BLANKS - fills the interrogate buffer with blanks from the end of message
            till the end of the buffer, and writes on tape.

            Input parameters: Register A - Blanks code.
                               Register Q - Serial No. of the first word to
                               be blank.

359

DOCUMENT CLASS ___DMPFMT___

PROGRAM NAME_____Dump Format From Disk To Tape_____

       SERCOR - requests core for the program buffers.

            Input parameters: Register Q - No. of requested words including header.

       IONOF - sets and clears I/O bits.

            Input parameters: Register Q - Backwards pointer of the buffer.

       BINASC - converts binary numbers to ASCII characters and stores them in STEMP.

            Input parameters: Register Q - The number to be converted.
                               STEMP   - Place to be stored after conversion.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    576 locations = 240_{16}

    There are 3 requests to CAM for memory spaces:

    a. 35 + 3 words for directory buffers.

    b. 192 + 3 words for LFD.

    c. Up to 1536 + 3 words for big buffer.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATIONS

    Re-entrant, unlocked.

DOCUMENT CLASS ___DMPFMT___

PROGRAM NAME ___Dump Format From Disk to Tape___

ANALYTICAL SUPPLEMENT

Buffers structure:

1. TAPE DIRECTORY

        size: 36 + 3 words

H
D
R
First format no. in the big buffer.
Last format no. in the big buffer.
First format no.
Format length
Second format no.
Format length
.   .   .
Last format no.
Format length

                                                34
                                                35

If there are less than 17 formats in system, zero word will signify the end
of the directory.

Last format no.
Format length
0

            35

DOCUMENT CLASS ___DMPFMT___

PROGRAM NAME___Dump Format From Disk To Tape___

2. FORMATS BIG BUFFER up to 1536 words

H
D
R
Format copied from
Disk/second format copied, etc.

1536

3. TAPE'S HEADER

DMPF   MMDDYY   XXXX

where: MM   - month no.
       DD   - day
       YY   - year
       XXXX - Big buffers size used for current dump (in words).

4. END OF DATA

EOD1   NNNN   XXXX

where: NNNN - last format no.
       XXXX - Big buffers size used for last dump (in words).

362

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME__DMT - Delete Mag-Tape Format__   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    Instruction syntax: DMT,nnn

    nnn is the Mag-Tape Format number to be deleted.
    Entered by the supervisor when that Format is obsolete.

2. DETAILED DESCRIPTION

    As all supervisor functions, DMT starts by making itself movable (through
    the SETMØV routine). Then, if no comma is found following 'DMT', an error
    is sent and the command rejected. Also, if nnn is illegal (nonnumeric),
    missing, or not between the allowed limits (1 to 255), the command is
    rejected.

    If nnn is legal, word 3 of MTFBUF is cleared (marking the format as inactive).
    Both the disk address and the core address of that format table are
    calculated and then the (inactive) Mag-Tape Format is written in its
    corresponding place on the disk, by the WRTDWA routine.. Control is then
    returned to SV.

INTERFACE

1. ENTRY METHOD

    DMT is called by DSKSUP through ENTPRG.

2. INPUT PARAMETERS

    (I) = base address of ATT.

3. EXIT METHOD

    If any errors detected, exit to REJSUP. Else exit to SV.

4. OUTPUT PARAMETERS

    Corresponding format on disk deactivated.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___DMT - Delete Mag-Tape Format___ Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ONEBIT ($23)
      ZROBIT ($33)
      ZERO ($22)
      SST ($47)

   b. SST EQUATES

      STEMP (20)
      EXTSST (0)
      MTFADD (42)
      SUBCOM (36)
      MTFBUF (46)
      MTFN (52)

   c. TRANSFER VECTOR

      GETBIN ($99)
      GETALP ($9B)
      REJSUP ($B2)
      SV ($B1)
      CLCDWA ($90)
      REDDWA ($A9)
      WRTDWA ($AA)

   d. EXTERNALS

      SETMOV

2. DATA AREAS USED

   ATT
   SST
   EST
   Interrogate buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $2A_{16} = 42_{10}$ words.

2. RELOCATION TYPE

   Loader relocatable, run anywhere.

3. REENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS _____ CYBERDATA _____

PROGRAM NAME_____ DMYFMT - Dummy Program for Mag-Tape User Routines_____ Level 06 _____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    This function is called if a nonexistent mag. tape Own Code, Label
    routine, or conversion routine is invoked. It points an appropriate
    message, and returns to the error entry point of MATER.

2. DETAILED DESCRIPTION

    The program checks whether the routine was invoked because of an
    undefined labelling routine, or a nonexistent Own Code. If JIDX is
    nonzero, MTFEXT indicates return to MTFMON, and MTFENT is -1, pprint
    "INVALID MAG. TAPE OCD NO." Else print "INVALID TAPE LABEL NO". Release
    the program via EXDFUN, and exit to the fourth entry point of MATER
    (via ENTPRG).

INTERFACE

1. ENTRY METHOD

    Invoked by ENTPRG when an undefined magnetic tape support routine
    is called. It needs four entry points defined, since a label routine
    has four entry points.

2. INPUT PARAMETERS

    The Supervisor Communication Area contains the information placed there
    by MATER or MTFMON.

3. EXIT METHOD

    Exit to the error exit of MATER via the following code:

    EQU   MATER ($805)

    LDQ   = XMATER

    ENA   3

    RTJ -   (ENTPRG)

4. OUTPUT PARAMETERS

    A message is printed on the supervisor list unit.

DOCUMENT CLASS _____CYBERDATA_____

PROGRAM NAME_____DMYFMT - Dummy Program for Mag-Tape User Routines_____Level 06_____

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. ATT

        JIDX   (3)

        MTFENT   (46)

        MTFEXT   (47)

    b. TRANSFER VECTOR

        SUPRW   ($97)

        ENTPRG   ($9E)

        EXDFUN   ($9F)

    c. GENERAL

        FW   (3)

        MTFMON   ($080F)

        MATER   ($0805)

2. MACROS

    None

3. PROGRAM DEPENDENCIES

    SUPRW

    EXDFUN

    ENTPRG

4. EXTERNALS

    None

DOCUMENT CLASS _____CYBERDATA_____

PROGRAM NAME_____DMYFMT - Dummy Program for Mag-Tape User Routines_____Level 06___

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $3A_{16} = 58_{10}$ program length.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS _____ CYBERDATA _____

PROGRAM NAME_____ DMYINT - Dummy Interrogate Function _____ Level 06 _____

PROGRAM FUNCTION

1. DESCRIPTION

   This program is called when a valid interrogate function is unpatched. The
   routine sets ATT/SW1/8 (Regen needed with RESET), and exits to ERROR with
   Q = $1B (Illegal Request).

INTERFACE

1. ENTRY METHOD

   Invoked by ENTPRG if an interrogate function disk address is set at $7FFF,
   at its only entry point DISTRT.

2. INPUT PARAMETERS

   I < current ATT base address

3. EXIT METHOD

   To ERROR with Q = $1B

4. OUTPUT PARAMETERS

   ATT/SW1/8 = 1
   I = current ATT base address

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      ONEBIT   ($23)

      ZROBIT   ($33)

   b. ATT

      SW1   (19)

   c. TRANSFER VECTOR

      ERROR   ($96)

d. GENERAL

    E1   ($1B)

2. MACROS

    None

3. PROGRAM DEPENDENCIES

    None

4. EXTERNALS

    None

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $B_{16} = 11_{10}$ program length.

2. LOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Dormant reentrant.

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    Called by DOC key to check the validity of the key and to get 3 characters
    If DOC key is illegal, exit to error. If the number is not 1-255 exit to
    error also. Also use to advance to next document in READ mode.

2. DETAILED DESCRIPTION

    DOC first calls SETMOV to clear the program header I/O bit. If mode is not
    Entry and at beginning of record, or Read, exit to error: Illegal Function
    Key. In case of Entry mode: set data type (ATT 13) to upper; jump to
    'DISPLAY' subroutine which displays 'D' character to signify that the following
    digits are DOC number.

    Get DOC number via GETSTR and store at the ATT. If CANCEL was entered,
    clear the first data line of the display, regenerate the parameters for
    the AID line, and exit via CLRSR. If an overflow condition occurred, or
    a number larger than 255 was entered, clear all format and document pointers,
    turn on SW1B10 and SW1B3, clear the document and format number displays,
    and exit to error (D3). If the number entered is valid, release this part
    of the program by 'EXDFUN' and enter the second part of REDOC by 'ENTPRG'.
    In READ mode: Clear flag word, if current record not written already,
    write the record on disk. If the basic mode is Verify check for End of Data
    through the bit in the TRB header; in Entry check for end of entered data.
    If end of data, set flag (ATEMP+4) to FFFF in VERIFY, and to 0 in ENTRY.
    If not end of Read (Entry) data and not one before, increment number of
    records in Read (RCDRD) and read the next record. If not the first in DOC.
    check if the format number of this record is different from the format
    number in ATT. If different, jump to subroutine 'GFCORE' (Get Format),
    set parameters to beginning of record and exit according flag. If flag
    positive, jump to 'DISPLAY' subroutine and exit via 'CLRSR' or 'ERROR'.
    If flag negative, jump to subroutine 'EXDFUN' and then exit via 'SETNFM'.

INTERFACE

1. ENTRY METHOD

    DOC is loaded via ENTPRG when the operator hits the DOC key.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.
    In the case of DOC SEL the input parameters are 1 up to 3 decimal digit 1-255.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __DOC__   Level 6

3. EXIT METHOD

    JMP   - (ERROR) When error occurs, except if in READ or VERIFY.
    JMP   - (CLRSR) After execution of function.
    JMP+   SETNFM   When end of data in Verify is reached during the Search.

4. OUTPUT PARAMETERS OR CONDITIONS

    None.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        SST ($47)
        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)

    b. SST

        STEMP (20)
        CFFA (12)
        READN ($A6)
        DISPLY ($A0)
        GFCORE ($9D)
        REWRIT ($A4)
        RELFOR ($AD)
        ERROR ($96)
        DISP ($A0)
        GETSTR ($A2)
        CLRSR ($93)
        EXDFUN ($9F)
        ENTPRG ($9E)
        GETBIN ($99)

    c. ATT

        BRCNT (15)
        RFA (14)
        FMTNO (13)
        BMOD (15)
        RCDRD ($31)
        RCD ($32)

           CFFA (12)
           ATRB (6)
           PFBA (7)
           FCNT (3)
           CURDA (9)
           RCNT (4)
           DT (13)
           CURCH (2)
           AINT (27)
           DOCNO (17)
           ATEMP (73)
           SW1 (19)

     d. ERROR CODES

        ERRB2 (7)   END OF DATA
        CA4 (4)   ILLEGAL FUNCTION KEY
        CD3 ($7)   INVALID NUMBER

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    ENEXSR -   (SETMOV - clear the program header I/O bit)
             -    (CLRSR - clear exit subroutine)
    ENTPRG -   Enter A Disk Resident Program
             -    (EXDFUN - exit Disk Resident Program)
    GSE   -   (ERROR - Display error at terminal)
    PIO   -   (READN - Read next record)
             -    (REWRIT - Write current record)
             -    (GETSTR - Get string of characters)
             -    (SETNFM - Set format motion bit 22 SW1)
    GFCORE -   Get format
             -    (RELFOR - Release FBA or TRB area)
    SCANV -   (GETBIN - Get the number in binary from input string)
    EORENT -   (FIRSTF - Set up tutorial parameters)
    EMM   -   (STTYPE - Set up data type in ATT)
    DISPLAY -   Handle display output
             -    (CLRFLD - Clear field number display)
             -    (DCHAR - Display D character)
             -    (DBOF - Display data type char. as before)
             -    (DFNDTY - Display data type char. upper U)
             -    (REGEN -
             -    (DDOC - Clear doc. display)
             -    (DFMT - Clear format display)

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____DOC_____ Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    AA_{16} = 170 words.

2. RELOCATION TYPE

    Run anywhere, disk resident.

3. REENTRANCY CLASSIFICATION

    Reentrant.

AA5641

PRINTED IN THE USA

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___DOCBCK - Document Back_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   Routine to process DOCBCK key in Read mode.

2. DETAILED DESCRIPTION

   DOCBCK first calls SETMOV to clear the program header I/O bit. Next it
   checks if Read mode. If not, illegal function error is generated. In
   case of Read mode reset flag (ATEMP+4). Then the program checks if it
   is the first record in batch. If beginning of batch set flag = FFFF. If
   not check SW1. If Bit 13=1 calls (REWRITE) to write the last record on
   disk, and resets bit 13. Since bit 13=0 reads the previous record from
   disk and decrements 1 from number of records in Read mode. The program
   loops backwards through the records looking for first in document. In
   case of first record in document the program checks if 'format number' in
   the ATT = 'format number' in TRB. If not 'GFCORE' is called to get new
   format, and then reset number of characters in field, number of characters
   in record. Relative FMT field address = 4. Add 4 to base address of
   format number and store in SST (current format field address). Store new
   data type in the ATT (STTYPE). Perform display and exit according to the
   flag situation:

   > FLAG $\geq$ 0 out via CLRSR
   > FLAG $<$ 0 out via error with the message
   > B1 (Beginning of Data).

INTERFACE

1. ENTRY METHOD

   DOCBCK: is loaded via ENTPRG when the operator bits $\overleftarrow{DOC}$.
            Execution begins at the only entry point $\overrightarrow{DOCBCK}$.

2. EXIT METHOD

   JMP - (ERROR) - When not in Read mode or beginning of data.
   JMP - (CLRSR) - When function terminates normally.

3. OUTPUT PARAMETERS

   None.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __DOCBCK - Document Back__   __Level 6__

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)

    b. SST

        CFFA (12)
        REWRITE ($A4)
        GFCORE ($9D)
        READP ($A5)
        DISPLAY ($A0)
        ERROR ($96)
        CLRSR ($93)

    c. ATT

        CMODE (2)
        FCNT (3)
        RCNT (4)
        ATRB (6)
        PFBA (7)
        SW1 (19)
        FMTNO (13)
        RFA (14)
        ATEMP (73)
        RCDRD (31)

2. ERROR CODE

    B_1 = Beginning of data.
    B_4 = Illegal Function key.

3. MACROS USED

    None.

4. PROGRAM DEPENDENCIES

    PIO   - (REWRIT - Write record on disk)
               (READP - Read previous record)
    GFCORE   - Get Format
    DISPLAY - Handle display output
               (DBSREL -
    GSE   - (ERROR - Display error at Terminal)
    ENEXSR   - (CLRSR - clears the exits stake Terminal bit, Function
               Lock and release Disk function, INTERROGATE Buffer)
               (SETMOV - clears the program header I/O bit)
    EMM   - (STTYPE - set up data type in ATT)

PHYSICAL CHARATERISTICS

1. LENGTH

    $3E_{16} = 62$ words.

2. RELOCATION TYPE

    Run anywhere, disk resident.

3. ENTRANCY CLASSIFICATION

    Reentrant.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME_____ DPR - Delete User's Program   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    DPR is a supervisor function to delete from the disk a user written data
    check own code program or a user written magnetic tape program of which
    there are three types: own code, tape label, and code conversion.

2. DETAILED DESCRIPTION

    DPR receives control from DSKSUP after storing the DPR command in the
    interrogate buffer. After calling SETMOV to enable itself to be moved,
    DPR scans and checks the command for syntax errors. It stores the 1st
    parameter (program number or directory index) in ATT/$2E in the case of
    an own code (2nd parameter 0), it checks the own code number to be not
    greater than maximum own code number in EST/4/8-15. In the case of a
    magnetic tape program (2nd parameter T) it checks that the program number
    is within the maximum for the type specified. It should be noted here
    that the area of the PRD reserved for mag-tapes entries is divided into
    3 sections. Words 0-31 are reserved for mag-tape own codes words 32-63
    for tape label routines and words 64-68 for code conversion programs.
    In cases of error DPR jumps to REJSUP with the appropriate error code
    in A. During this initial phase DPR calculates the ordinal of the
    entry in the PRD depending on the parameters and stores it in ATT/$2E
    Subroutine CORBIT is called to locate the corresponding status bit in
    the PRD control block (words 8 onwards of PRD).

    If the program is currently not in core a check (status bit = 0) is made to
    see if the PRD entry equals 7FFF, indicating an invalid PRD address, and
    if so DPR jumps to REJSUP with A = 13. If the PRD entry is valid
    (i.e., ≠ 7FFF) it and the corresponding entry in the disk core image
    are deleted by setting them to 7FFF.

    If the program is in core (status bit = 1) then the core address in the
    PRD entry is used to get the user count from word 2 of the program header.
    If the user count is nonzero, the program busy error is indicated via a
    jump to REJSUP. If user count is zero the status bit in PRD control block
    is cleared and PRD entry in core and in the disk core image are deleted
    by setting them to 7FFF. Exit is to SV.

SUBROUTINE CORBIT (locate status bit in PRD control block)

CORBIT subtracts the distance to the miscellaneous entries (PRD/0) from the
ordinal in the PRD plus one (ordinal is in ATT/2E) to get the bit position
in the PRD control block.

Dividing by 16 gives the word and the bit position in this word of the status
bit corresponding to the PRD entry. CORBIT returns with A = word in PRD
control block and Q = bit position in this word.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME_____ DPR - Delete User's Program   Level 6

INTERFACE

1. ENTRY METHOD

    DPR is a disk resident program which is called via ENTPRG.

2. INPUT PARAMETERS

    DPR command as entered by supervisor is stored in IRB buffer attached to
    supervisor ATT. I = current ATT address.

3. EXIT METHOD

    Error exit by jump to REJSUP with A appropriate error code. Normal exit
    by jump to SV.

4. OUTPUT PARAMETERS OR CONDITIONS

    On normal exit the PRD entry and its corresponding entry in the core image
    on disk are set to 7FFF and the corresponding status bit in the PRD control
    block is cleared.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)

    b. SST

        PRD (3)
        STEMP (20)
        CLCDWA ($90)
        GETBIN ($99)
        GETALP ($9B)
        GETCOM ($9C)
        WRTDWA ($AA)
        SV ($B1)
        REJSUP ($B2)

    c. EST

        PRDDA (10)

    d. ATT (SCA)

        ATEMP (73)
        DIRIDX (46)

e. CONSTANT ALPHA CHARACTERS

    C ($43)
    L ($4C)
    O ($4F)
    T ($54)

2. PROGRAM DEPENDENCIES

PIO	CLCDWA	Calculate disk word address
	WRTDWA	Write disk word address
SCANV	GETBIN	Get characters up to comma and convert to binary.
	GETALP	Get next alpha character from input string
	GETCOM	Get comma, EOR, or nonspace character from input string.
SUPER	REJSUP	Supervisor reject procedure
	SV	Supervisor command processor
ENEXSR	SETMOV	Enable disk resident program to move

3. DATA AREAS USED

    ATT   Active terminal
    SCA   Supervisor communication area
    PRD   Program directory

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $7B_{16} = 123_{10}$

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __DRC - Display Records in a Packed Format_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    DRC is an interrogate function called in read mode that displays data on
    the 970-480 keystation starting with the current field, in a packed fcrmat:
    Fields are separated by inverse blanks (error - flagged fields are prededed
    by an inverse asterisk). A new record is begun on a new line.

    The 8 data lines are filled; a CANCEL or CONTINUE key is not required.
    CANCEL regenerates the normal display at the last field reached in the
    packed format. CONTINUE results in another full screen of display. If
    CONTINUE is pressed at end-of-data, the normal display is regenerated and
    a B2 (end-of-data) error is displayed. Display is performed by using the
    DSPDIR call, which allows sending a 5-word call to the CDATA CRT display
    module from outside the display driver routines.

2. DETAILED DESCRIPTION

    DRC: SETMOV is called to allow the program to be moved by MMM. SST + STEMP
    is set to 3 to point to the third (counting from zero) position in the IRB.
    GETCOM is called to check for end-of-record; the DRC request has no parameters.
    If, on return Q is positive, indicating end-of-record was not detected,
    control goes to ERRE1.

    ATT/CMOD/8, 12, 13 are checked to be zero (read mode and not pseudo-entry
    and not pseudo-verify). If so, control goes to OKMODE.

    a. ERRE1:   ATT/SW1/8 is set to 1 (regen required) and the program exits
        to ERROR, with Q=code for E1 (invalid request).

    b. OKMODE: If ATT/Word 0/10 = 0 go to ERRE1 (station type is self-scan)
        ATT/RCNT is set to ATT/RCNT-ATT/FCNT/0-7.

        (The record column count is set to the beginning of the current field.)

        ATT/FCNT/0-7 is set to zero.

    The display mode is changed from interrogate to data.

    a. NWSCRN: The screen is cleared and the header regenerated. The
        address of the beginning of the data area is converted to address
        codes and saved in CALL. The address of CALL is determined by an
        RIJ command just before CALL, which stores the address of CALL in
        REFADR. The address codes are passed to the display driver with
        a call to DISPLY (DSPDIR entry).

b. CHK01: End-of-data is checked:

1) Basic mode entry - If ATT/RCDRD = ATT/RCD and
   ATT/BRCNT/0-11 = ATT/RCNT
   then exit to ERRB2. Otherwise jump to NOTEND.

2) Basic mode verify - If this is the first field in the last record
   and ATT/CURDA, CURDA + 1 = previous address in
   the terminal record buffer, exit to ERRB2.
   Otherwise, jump to NOTEND.

c. ERRB2: Call CONCAN. If cancel was entered, jump to CANEXT.
Otherwise, set the display mode to data, regenerate the display and
exit to ERROR with code B2 (end-of-data).

d. NOTEND: Check that there are at least two display positions on the
screen: If UPDAT - TOPT/CURSRD is less than 2, jump to CONCAL.

Save the above difference in ATT/ATEMP. Get the first character of the
field using GBYTE.

If the field is error flagged, display an inverse asterisk, otherwise
display an inverse blank.

The display is performed by sending the following sequence of codes:
Display inverse ($1C), blank or asterisk ($20 or $2A), display normal
($1D).

TOPT/CURSRD is increased by 1.

The field length is taken from the format, saved in ATT/ATEMP + 1, and
compared to the value saved in ATT/ATEMP.

If the field is longer than the ATEMP value, the ATEMP value is taken
as a string length. Otherwise the field length is used.

A 5-word parameter block is built for a display call. The parameters
indicate the starting address and byte of the field, the starting and
final cursor position, and the address in the TRB where the I/0 bit
should be set.

The field is then displayed.

If ATT/ATEMP - ATT/ATEMP + 1 is negative, only a partial field was
displayed; jump to CHK01. Otherwise, the field description length
is isolated. (FMT/0/0-3). If this value is zero, the field just
displayed was the last in the record: jump to NXTRBC.

Format pointers are advanced to the next field (ATT/RFA, ATT/RCNT, and
SST/CFFA) and control goes to CHK01.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __DRC - Display Records in a Packed Format__   Level 6

    e.   (NXTREC). The current record is tested to see if it is the last:

        1)   Basic mode verify: if TRB/5/15 is 1 jump to ERRB2; otherwise jump to RWCHK.

        2)   Basic mode entry:
            If ATT/RCDRD = (ATT/RCD) - 1
            and ATT/BRCNT/0-11 = 0
            go to ERRB2.

    f.   (RWCHK). If a rewrite is required (ATT/SW1/13=1), call REWRIT. Call READN. Increment ATT/RCDRD.

        If the format of the new record does not equal ATT/FMTNO/0-13 call GFCORE to read the new format.

        Set: ATT/RCNT to 0
              ATT/RFA to 4
              SST/CFFA to point to the first field of the new record.
        If the cursor is at the beginning of a line, jump to CHK01.
        If this is not the eighth data line jump to NEWLIN.

    g.   (CONCAL). Call CONCAN. If continue was entered jump to NWSCRN.

    h.   (CANEXT). Set the display mode to data, regenerate the display, and exit to CLRSR.

    i.   (NEWLIN). Calculate the screen address of the next line, convert to display codes, transmit the codes and jump to CHK01.

INTERFACE

1. ENTRY METHOD

    DRC is called by INTRGT with a call to ENTPRG for interrogate function ENTPRG jumps to the first program instruction.

2. INPUT PARAMETERS OR CONDITIONS

    (I) = base address of active terminal table (ATT)
          ATT/AINT contains the address of an interrogate buffer.

3. EXIT METHOD

    a.   Jump to ERROR with either E1 code (invalid request) or B2 code (end of data).

    b.   Jump to CLRSR when the function has been cancelled.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___DRC - Display Records in a Packed Format_____Level 6

4. OUTPUT PARAMETERS OR CONDITIONS

    (I) = base address of ATT.
          When exiting on an error condition,
    Q   = error code index.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ONEBIT ($23)
        ZROBIT ($33)
        ZERO ($12)
        SST ($47)
        CLRSR ($93)
        GBYTE ($94)
        ERROR ($96)
        GETCOM ($9C)
        GFCORE ($9D)
        DISPLY ($A0)
        REWRIT ($A4)
        READN ($A6)

    b. SST

        CFFA (12)
        STEMP (20)

    c. ATT

        CMOD (2)
        FCNT (3)
        RCNT (4)
        ATRB (6)
        PFBA (7)
        CURDA (9)
        FMTNO (13)
        FLON (14)
        RFA (14)
        BMOD (15)
        BRCNT (15)
        SW1 (19)
        RCDRD (31)
        RCD (32)
        TOPT (33)
        ATEMP (73)

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___DRC - Display Records in a Packed Format_____Level 6

    d. ERROR CODES

        B2 code ($7)
        E1 code ($12)

    e. MISCELLANEOUS

        CRSRD (3)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

    SETDIS - set display type
    CLRALL - clear display screen
    GENHDR - generate display header
    LOWDAT - address of beginning of data area on screen
    CONCSR - subroutine to convert from screen address to output address codes
    DSPDIR - send a 5 word control block to the subroutine CDATA
    CARLIN - no. of characters per display line
    UPDAT   - address of last data display location
    CONCAN - subroutine that waits until either cancel or continue code is entered
    SETMOV - allows current disk resident program to be moved by MMM
    REGEN   - regenerate the display according to current table conditions
    GBYTE   - get the current character from the terminal record buffer
    GETCOM - scans input message for end of record
    GFCORE - reads current format into memory
    DISPLY - central display call. Used with the following parameters:

    REWRIT - rewrites the current record to disk
    CLRSR   - clears enter/exit stack and function locks
    READN   - reads in the next record from disk

4. DATA AREAS USED

    SST
    ATT
    TRB
    FMT

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME__ DSKER1 - Disk Error Advisory (1)__

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   DSKER1 tries to make more core available for the main overlay of the advisory program.

2. DETAILED DESCRIPTION

   DSKER1 is loaded by PIO after an erroneous disk I/O has occurred. All the buffers whose user count is nonzero and which point to the station's ATT are checked. If the corresponding word in the ATT also points to the buffer, they are released.

   The words TRB, AINT, ASRCH (6, 27, 28 in ATT) which usually point to buffers are cleared. The interrogate bit is cleared. Monitor override bit is set.

   The program tries to load the next overlay DSKER2. If no core is available it exits to error exit in PIO, ASK17. Otherwise the ASKER2 overlay is executed.

INTERFACE

1. ENTRY METHOD

   PIO attempts to load the program via ENTPRG whenever it detects an error in disk I/O. If the program is loaded, then it is executed.

   ```
   CLR     A
   LDQ  =  N$8014  Miscellaneous program 14; load only
   RTJ  -  (ENTPRG)
   SQZ     DSK17  no core available
   ADQ  -  5, Q    first entry point
   JMP  -  1, Q    execute
   DSK17:  prints "no core for recovery program"
   ```

2. INPUT PARAMETERS OR CONDITIONS

   I = current ATT base address

   (ATEMP, I) - temporary location in the ATT (word 73) contains the sector address in the erroneous request.

(CURCH, 1) - word 2 in ATT, 8 L.S.B. contain the disk number in erro-
neous request. BIT 7 is set if the error is due to "not
ready" state or nonexistent disk.

3. EXIT METHOD

    a. If enough core is available the main (second) overlay DSKER2 is loaded.

```
CLR     A
LDQ  =  N$8015  miscellaneous program 15; load only
RTJ  -  (ENTPRG)
SQZ     NOCORE
ADQ  -  5, Q
JMP  -  1, Q
NOCORE
```

    b. If no core available for second overlay, program exits to error exit
in PIO, DSK17

```
JMP+   DSK17
```

4. OUTPUT PARAMETERS OR CONDITIONS

ATEMP, 1 - same as input parameter.

CURCH, 1 - same as input parameter.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

NZERO	($12)
ZROBIT	($33)
ONEBIT	($23)
ZERO	($22)
LPMASK	(2)

    b. SST

SST	($47)
ATTL	(13)

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___DSKER1   -   Disk Error Advisory (1)_____ Level 06

	FRPLL	(6)
	FRPP	(8)
	STEMP	(20)

c. ATT

	CMODE	(2)
	BMOD	(15)
	SW1	(19)
	SW2	(20)
	FUNIDX	(21)
	FMTNO	(13)
	PFBA	(7)
	AINT	(27)
	ATEMP	(73)
	BUFFER	(ATEMP+2)

d. TRANSFER VECTOR

	ENTPRG	($9E)
	EKDFUN	($9F)
	RELFOR	($AD)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

PIO	DSK17
	ENTPRG
ENTPRG	EXDFUN
ENEXSR	SETMOV
GFCORE	RELFOR

4. DATA AREAS USED

ATT

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ DSKER1 - Disk Error Advisory (1)   Level 06___

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$40_{16} = 64_{10}$

2. RELOCATION TYPE

Run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___DSKER2 - Disk Error Advisory (2)_____Level 06_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    A check is made to ascertain which area on the disk has a faulty track. If
    the error occurred in the data area, the corresponding batch is put to the
    "frozen" state (if it is not frozen yet), or is deleted if the entered batch
    is too short or the disk I/O error is irrecoverable. If the error occurred
    in the system tables and the station is working on an active batch, its batch
    is frozen or deleted.

    The station which performed the I/O receives the message "disk I/O error"
    and will remain in the idle mode, and under supervisor lock. If the disk
    address in the faulty I/O request belongs to the allocatable disk's area,
    the corresponding track is logged in special system table BTD (Bad Tracks
    Directory) in DAM.

    Informatory messages about the erroneous track, the area on disk to which
    the track belongs and the status of the erroneous batch are printed on the
    primary supervisor station.

2. DETAILED DESCRIPTION

    The list logical unit in the supervisor communication area is assigned the
    value of the comment device logical unit; thus SUPRW will write the inform-
    atory messages through the MESSAG subroutine on the primary supervisor logical
    unit (usually TTY).

    If the disk error detected was "not read" (BIT 7 in CURCH, 1 is set) the pro-
    gram branches to the common exit. A message with the erroneous sector address,
    the disk number and the number of the station which performed the I/O is
    written on the TTY.

    A check is made in which disk area the error occurred. The disk number is
    transferred in 8 L.S.B. of (CURCH, 1), sector address in (ATEMP, 1). If the
    disk number is not zero, it was obviously data. If the sector address belongs
    to the allocatable area of the system disk it can be either format area or
    data. In those cases the bad track (track containing the sector) is logged
    in the Bad Track Directory (BTD) in DAM. The least significant bits in the
    track address include the disk number 0 - 3.

    If the error occurred in the allocatable area of the system disk, the FMTLNK
    (Format Link Table) is read (if it is not empty) into core and searched. If
    the erroneous track is found among the tracks listed in it, the LFD is read

into memory by sectors and searched until the format in which the error
occurred is found. The format inactive bit in the LFD is set and is written
back to disk. A message informing the supervisor of the erroneous format
is printed. If the track is not found in the FMTLNK it is probably data.

When the erroneous sector in the system disk does not belong to data or
format area, it is compared to the beginning of scratch (value of core
location $C2) and the end of scratch (system install parameter SECTOR).
If the sector belongs to the scratch area, an informatory message is printed.
If the sector is not part of the scratch area it should be part of the system
tables. A check is made to find out to which table the sector belongs, and
an informatory message is printed.

If the error did not occur in the data area, but the station which performed
the I/O is in Entry or Verify mode it is verified that the job index and
batch number in the ATT corresponds to a legal ABT entry and the batch is
handled as an erroneous batch. If the error occurred in data storage, an
informing message is printed. If the I/O request was issued by a station
in Entry or Verify mode, the batch connected is the first to be suspected.
FTOB (First Track of Batch) in ATT and tracks listed in TKT (Track Table in
Sector 0 in First Track of Batch) are compared to the erroneous track. If a
match is found it is verified that the job index and the batch number in the
ATT correspond to a legal ABT entry, whose track address is equal to FTOB in
ATT. If the tests fail, or if an error occurred in data storage but the
station is not in Entry or Verify mode, the ABT's are searched. The erroneous
track is compared to the first track of batch in the ABT entry and the tracks
listed in the TKT (Tracks Table) until the batch containing the erroneous
track is found. If the disk I/O error occurred in the ATT image or the TKT,
the ABT entry is cleared. If the error is not in the TKT, the tracks of the
batch are released. If the batch is in maintenance mode, it receives its
original mode. If the batch is active it receives the following treatment:
The ATT image in first track of batch is read into memory.

If the batch is in entry mode and has less than 16 records, it is deleted;
otherwise, the batch is frozen. If the ATT is of a "frozen" state (RCNT
less than 0) no modification of ATT is needed. Else, the monitor check
override bit is set, accumulated time is updated, and the ELDT (entry in
LDT) is kept in relative mode. Backwards pointers to nonexistent buffers
(AINT, ASRCN, TRB) are cleared. The modified ATT is written back to disk.
The ABT entry is modified to frozen.

Informing message on the erroneous batches status is printed.

Common Exit: The station which performed the erroneous I/O will be assigned
the idle state and will remain under supervisor lock. .The message "DISK I/O
ERROR" will be displayed on the screen.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___DSKER2 - Disk Error Advisory (2)_____Level 06____

INTERFACE

1. ENTRY METHOD

    DSKER1 tries to load the program via ENTPRG. If there is enough memory
    available, the program is loaded and executed.

```
        CLR   A
        LDQ = N$8015    Load only, miscellaneous prg 15.
        RTJ - (ENTPRG)
        SQZ   NOCORE    no core available
        ADQ - 5, Q      first entry point
        JMP - 1, Q      execute
      NOCORE
```

2. INPUT PARAMETERS OR CONDITIONS

    1 = current ATT base address

    (ATEMP, 1) - temporary location in the ATT (word 73) contains the sector
                 address in the erroneous request.

    (CURCH, 1) - 8 L.S.B. contains the disk number. Bit 7 is set if the error is
                 due to "not ready" state, or nonexistent disk.

3. EXIT METHOD

    The program exits to the dispatcher via CLRSR.

        JMP - (CLRSR)

    If there is not enough core for memory buffers the program releases itself
    and exits via error exit DSK17 in PIO with the message 'DISK I/O ERROR'. NO
    CORE FOR RECOVERY PGM. The station remains under function lock.

4. OUTPUT PARAMETERS OR CONDITIONS

    a. A message including the disk number, the erroneous sector address and the
        station number will be printed on the Teletype.

            ERROR IN DISK nn, SECTOR mmmm, STATION 11.

            where: nn - disk number
                    mmmm - sector address
                      11 - station number

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___DSKER2 - Disk Error Advisory (2)_____Level 06_____

b. Detecting the area in which the disk I/O error occurred, the following informing messages can be printed on the teletype:

    ERROR OCCURRED IN SYSTEM TABLES

    ERROR OCCURRED IN STATION TYPE TABLE

    ERROR OCCURRED IN PROGRAM TABLE

    ERROR OCCURRED IN FORMAT TABLE

    ERROR OCCURRED IN JOB TABLE

    ERROR OCCURRED IN DOCUMENT TABLE

    ERROR OCCURRED IN BATCH DIRECTORY TABLE

    ERROR OCCURRED IN ERROR MESSAGE TABLE

    ERROR OCCURRED IN MAG TAPE FORMAT TABLE

    ERROR OCCURRED IN SQUEEZE TABLE

    ERROR OCCURRED IN TAPE RECOVERY TABLE

    ERROR OCCURRED IN STATISTICS TABLE

    ERROR OCCURRED IN DISK ALLOCATION TABLE

    ERROR OCCURRED IN CHECK DIGIT WTS TABLE

    ERROR OCCURRED IN SCRATCH AREA

    ERROR OCCURRED IN FORMAT nnnn

        where: nnnn - format number

    ERROR OCCURRED IN DATA STORAGE

    JOB jjjjjj, BATCH bbb

    JOB jjjjjj, BATCH bbb FROZEN

    JOB jjjjjj, BATCH bbb DELETED

        where: jjjjjj - job name

                  bbb - batch number

c. The message DISK I/O ERROR will be displayed on the screen of the station whose request was erroneous.

d. The station remains under supervisor lock.

AA5641

PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ DSKER2 — Disk Error Advisory (2)   Level 06___

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK   ($2)

        NZERO   ($12)

        ZROBIT   ($33)

        ONEBIT   ($23)

        ZERO   ($22)

        TEN   ($46)

    b. SST

        SST   ($47)

        STEMP   (20)

        CLOCKS   (17)

    c. EST

        PRDDA   (10)

        SPIL   (23)

        LDDA   (3)

        MAXFMT   (24)

    d. TRANSFER VECTOR

        CLCDWA   ($90)

        ENTSR   ($91)

        EXTSR   ($92)

        CLRSR   ($93)

        SUPRW   ($97)

        RTNSR   ($98)

        REDSEC   ($A7)

        WRTSEC   ($A8)

        REDDWA   ($A9)

        WRTDWA   ($AA)

AA5641   PRINTED IN THE USA

RELFOR	($AD)		
CAM	($AE)		
DISP	($A0)		
SPILL	($A1)		
WAIT	($B0)		
EXDFUN	($9F)		

e. ATT

FTOB	(1)		
CURCH	(2)	-	8 L.S.B. holds the disk number.
PFBA	(7)		
RCNT	(4)		
JIDX	(3)		
CMODE	(2)		
TRB	(6)		
ELDT	(16)		
DOCNO	(17)		
BMOD	(15)		
SW1	(19)		
SW2	(20)		
ABTIDX	(22)		
AINT	(27)		
ASRCH	(28)		
ECODE	(34)		
BATCH	(35)		
ABTBUF	(46)	-	pointer to ABT buffer header.
ABTPTR	(47)	-	pointer to current entry in ABT buffer.
ABTDA	(48)	-	current sector address of ABT page to be read.
TRKS	(49)	-	pointer to TKT core buffer.
ATEMP	(73)	-	contain the faulty sector address.
ABTWD1	(50)	-	first word in ABT entry (status + job index).
RCD	(32)		

FLG   (51) - internal flag.

TRACK (ATEMP + 1) - faulty track address.

BUFFER (52) - backwards pointer to core buffer.

SUPCOM (36)

LSTLUN (SUPCOM + 1)

COUNTE (54)

SAVEQ   (53)

f. GENERAL EQUATES

FW   (3)

BEGSCH ($C1)   - beginning of scratch.

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

        CLCDWA

        SUPRW

PIO   REDSEC

        WRTSEC

        REDDWA

        GETBUF

        DSK17 - Error exit no core for recovery program.

        ENTSR

        EXTSR

ENEXSR RTNSR

        CLRSR

        SETMOV

        WAIT

CAM

GFCORE RELFOR

SUPER SUPTHD   - Thread of supervisor read request from TTY.

DISPLAY SETDIS - Set display mode.

        CLRALL   - Clear the screen.

BORENT   LINKTR - Release track listed in first track of batch when deleting the batch.

GETLJD

ARITH   BD

DAM   BADTRK - Bad tracks directory.

SYSTEM INSTALL PARAMETERS, DISK ADDRESSES

SECTOR - end of scratch area of disk

SIBSEC - beginning of allocatable disk

SIESEC - end of allocatable disk

SIFLNK - directory of tracks allocated for formats

SILFD   - formats directory

SIABTW - length of active batches directory in words

SITYPE - stations type table

SIPRDN - area reserved for programs

SILJD   - legal job directory

SILDD   - legal document directory

SISPIL - active batches table

SIERRM - error messages table

SIDMTF - magnetic tape formats table

SIABTS - length of active batches directory in sectors

SIDSQZ - squeeze work area

SITRDA - tape recovery area

SIMDMY - programs area

SISTDA - statistic buffer

SIDTR1 - track's table

SIDCWT - check digit weights table

SIREJM - reject messages table

EXTBV4 - extended system table

LOG1A   - logical units physical table directory

4. DATA AREA USED

BTD - Bad Tracks Directory in DAM

BTD configuration

```
┌─────────────────┐
│                 │
│   64 words      │
│                 │
├──────────┬──────┤
│ SEC ADR  │  LU  │      sector address + disk number 0-3
│          │      │
│                 │
│                 │
│        •        │
│        •        │
│        •        │
│        •        │
│        •        │
│        •        │
│        •        │
│                 │
│                 │
└─────────────────┘
```

ATT

ABT'S

LFD

FMTLNK

DTL - ATT image

LDD

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

460_{10} program length.

ABT page + 93 max buffer length.

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___DSKER2 - Disk Error Advisory (2)___   Level 06

2. RELOCATION TYPE

   Run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

AA5641   PRINTED IN THE USA

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___DSKSUP___

PROGRAM NAME___Disk-resident portion of Supervisor Command Processor___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   DSKSUP checks the current Supervisor Command for validity at the current
   terminal, and rejects an invalid command. For valid commands, DSKSUP
   initiates execution of the disk-resident function to process the Supervisor
   Command.

2. DETAILED DESCRIPTION

   a. (DSKSUP). Call SETMOV to enable DSKSUP to be moved, if necessary.

      ● If ATT/0/0-5 ≠ 0 (not terminal 0), go to step b.

      ● Clear i/0 bit in Interrogate buffer.

      ● If ATT/SVTMP1/4 = 1 (fewer words read than requested), set $FFFF
        in the word following the last word of data in the input buffer.

      ● Save the number of words read in the Supervisor write request.

      ● If the Supervisor input unit is the system input comment unit,
        go to step f.

      ● Call SUPRW to display the command, and go to step f.

   b. (CHKIN). Call DISPLY to clear screen.

   c. (CONT). If input word count is zero, go to step e.

   d. (REDSP). Call SUPRW to display input command.

   e. (CHGMOD). Call DISPLY to select idle area of screen for output.

   f. (CHKCMD). Call GETCH to input first 3 characters of command and pack
      them into one word in ATT/SVTMP1.

   g. (CK). Compare input command with entries in CODE table.

   h. (REJECT). If no match is found, set (A) = 33 and exit to REJSUP to
      issue the message UNDEFINED COMMAND and wait for another Supervisor
      command.

   i. (LOAD). If a matching command was found in the CODE table, clear
      ATT/SCA + 10 through ATT/SCA + 27 (Supervisor Command Area).

      ● Check the corresponding entry in the EPT table to determine whether
        the command is valid for the terminal:

1. If ATT/0/10 = 0 (Self Scan) and ATT/0/0-5 = 0 (Terminal 0) and EPT entry bit 15 = 1 (OK on Terminal 0) go to step j.

2. If ATT/0/10 = 0 (Self Scan) and EPT entry bit 12 = 1 (OK on Self Scan), go to step j.

3. If ATT/0/10 = 1 (CRT) and EPT entry bit 14 = 1 (OK on CRT), go to step j.

4. If ATT/0/10 = 1 (CRT) and EPT entry bit (c3) (OK on CRT with Supervisor authorization), and Terminal Type Table entry for this terminal has bit 7 = 1 (Supervisor authorization), go to step j.

5. Otherwise, set (A) = 34 and exit to REJSUP to issue the message CMD INVALID ON THIS STATION and wait for another Supervisor command.

j. (OK). Call EXDFUN to release DSKSUP program area.

● Call ENTPGM to load and execute the requested function. If sufficient storage is not available, ENTPGM exits to REJSUP with Memory Full code.

INTERFACE

1. ENTRY METHOD

DSKSUP is a disk-resident program which is called via ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

(I) = Base address of Active Terminal Table (ATT)

3. EXIT METHOD

Normal exit is to a command processor program via ENTPRG (ENTPGM).

Error exit is to REJSUP with (A) = error code.

4. OUTPUT PARAMETERS OR CONDITIONS

(I) = Base Address of ATT

ASSEMBLY CHARACTERISTICS

1. EQUATES

a. LOCORE

LPMASK ($2)
ZERO ($22)
ONEBIT ($23)
SST ($47)
CLCDWA ($90)

30

DOCUMENT CLASS ___DSKSUP___

PROGRAM NAME___Disk-resident portion of Supervisor Command Processor___

    SUPRW ($97)
    GETALP ($9B)
    ENTPGM ($9E)
    EXDFUN ($9F)
    DISPLY ($A0)
    REDDWA ($A9)
    REJSUP ($B2)

b. SST

    STEMP (20)

c. EST

    TTB (41)

d. ATT

    AINT (27)
    SUPIN (36)
    SCA (36)
    SVTMP1 (46)
    ATEMP (73)

e. ALPHA CHARACTERS

    A ($41)
    B ($42)
    C ($43)
    .
    .
    .
    Z ($5A)

f. EPT TABLE CONSTANTS

    TO (1)
    CR (1)
    CRSUP (1)
    SS (1)

g. OTHER CONSTANTS

    FW (3)
    SVSKEL ($3000)

2. MACROS USED

    CODE   TABLE   ENTRY

    CMD   MAC   A,B,C
            ADC   'A'*1024 + 'B'*32 + 'C' - $40*1024 - $40*32 - $40
            EMC

31

```
EPT     TABLE       ENTRY

IDX     MAC         TO,CR,CS,SS,IX,EP
        VFD         X1/'TO',X1/'CR',X1/'CS',X1/'SS',X4/0,X6/'IX',X2/'EP'
        EMC
```

3. PROGRAM DEPENDENCIES

$$
\text{PIO} \begin{cases} \text{CLCDWA} & \text{Calculate Disk Word Address} \\ \text{SUPRW} & \text{Supervisor Read/Write} \\ \text{REDDWA} & \text{Read Disk Word Address} \end{cases}
$$

SCANV   (GETALP) Get Next Alpha Character from Input String

$$
\text{ENTPRG} \begin{cases} \text{ENTPGM} & \text{Enter a Disk Resident Program} \\ \text{EXDFUN} & \text{Exit Disk Resident Program} \end{cases}
$$

DISPLY   Display a Message

SUPER   (REJSUP) Supervisor Reject Procedure

ENEXSR   (SETMOV) Enable Disk Resident Program Move

4. DATA AREAS USED

```
SST     System Status Table
EST     Extended Status Table
ATT     Active Terminal Table
TTB     Terminal Type Table
CODE    Supervisor Command Table
EPT     Command Validity/Entry Point Table
```

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $EA_{16} = 234_{10}$ words

2. RELOCATION TYPE

   Disk resident, run anywhere

3. ENTRANCY CLASSIFICATION

   Reentrant

32

DOCUMENT CLASS ___DST_____

PROGRAM NAME___Display Header for 970-32_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   This interrogate function displays the information contained in the CRT header
   at the 970-32. The format is JJJJJJ BBB RRRR DDD FFF where:

   JJJJJJ = Job name
   BBB   = Batch number
   RRRR   = Record count
   DDD   = Document number, if any
   FFF   = Field number

   DST is the only entry point.

2. DETAILED DESCRIPTION

   DST is an overlay loaded via ENTPRG. First DST calls SETMOV to clear the I/O
   bit in the program header so the program can be moved in core. Next GETLJD
   is called to get the job name from the legal job directory. The job name and
   all subsequent parameters are stored in the interrogate buffer. An internal
   subroutine is used to convert a three digit BCD number to ASCII for the batch
   number (stored in ATT as BCD), record count and document number, if any, and
   field number (converted to BCD via BD). DISPLY is called to set the fact
   that reset is needed for the next display regeneration. ATT/SW1/7,8 are set
   to indicate reset must regenerate the display and reset or disp key must be
   the next character. The ATT/CMODE/14 is set to IDK mode and exit is via CLRSR.

INTERFACE

1. ENTRY METHOD

   DST is a disk resident program loaded via ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

   I = current ATT base address.
   No special parameters are directly passed to it.

3. EXIT METHOD

   Exit to dispatcher after clearing exits stack via CLRSR.

4. OUTPUT PARAMETERS OR CONDITIONS

   I = current ATT base address.
   Terminal's interrogate buffer is filled with the header information.

35C

DOCUMENT CLASS ____DST_____

PROGRAM NAME_____Display Header for 970-32_____

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ONEBIT ($23)
        ZROBIT ($33)
        CLRSR ($93)
        DISPLY ($A0)

    b. ATT

        CMODE (2)
        FLDNO (14)
        DOCNO (17)
        SW1 (19)
        AINT (27)
        RCDRD (31)
        RCD (32)
        BATCH (35)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

    ENEXSR (SETMOV) - Clear I/O bit in header.
          (CLRSR) - Clear exits stack and go to dispatcher.
    DISPLY   - Set data definition bits.
    CRMTDS (DLINE) - Reset needed next regeneration constant.
    GETLJD   - Get job name from legal job directory.
    ARITH (BD)   - Binary to decimal conversion.

4. DATA AREAS USED

    ATT
    Interrogate buffer

351

DOCUMENT CLASS ___DST___

PROGRAM NAME___Display Header for 970-32___

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $GF_{16} = 111_{10}$ words

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___DUPKEY__

Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   DUPKEY processes the manual DUP function. The terminal must be in entry,
   pseudo entry for Insert record, or verify mode. Duplication from a
   record of the previous format is allowed if the previous record was long
   enough. DUPKEY is not allowed on the first record of the batch. The
   data to be duplicated must pass the same validation test as newly entered
   data. Exit is to EMM or VMM depending on the terminal basic mode. DUPKEY
   has a single entry point.

2. DETAILED DESCRIPTION

   DUPKEY first calls SETMOV to clear the program header I/O bit. Next
   terminal mode checks are made. Pseudo verify, and pseudo entry both with
   no Insert, and read modes generate ILLEGAL FUNCTION KEY via ERROR. An
   attempt to use the DUPKEY on the first record in the batch also generates
   an Illegal function error. If the record being entered is the first one in
   a format, the previous record is read into core via internal subroutine
   READ 1 to see if it is long enough. If not Illegal function error is
   generated. DUPKEY is not allowed in fill blank if numeric field, and blank
   appear, in this case internal subroutines FBFIQ checks it illegal function
   error generated. IN ENTRY mode each character of the record from which the
   duplication is being performed, is validated according to the data type of the
   field which is currently being entered.

   If all characters pass the validation check, the characters are moved one at
   a time from the previous record to the current record via SBYTE.

   IN VERIFY mode, the characters from the current record are verified against
   the characters from the previous record. A mismatch results in the terminal
   mode being changed to IDK, and the column number and mismatch error being
   displayed. When the entire field (or the remaining portion has been
   transferred), the manual SKIP/DUP field is displayed, the column number is
   displayed the program core is released via EXDFUN and exit is to VMM (END OF I)
   or EMM (EMM I) depending on the terminal mode.

   When DUP is depressed in ENTRY with no change in front, GBYTE is used to
   extract the first character of the field. A check is made for DUP in the
   middle of a numeric field where the previous held blanks (FBFIR). If this was
   the case, exit is to ERROR. Otherwise, the character extracted is stored
   back in the TRB via SBYTE, without the error flag bit.

DUP IT is worked to a durance pointer to end of field, and to set up display parameters. Exit is to EMM1.

INTERNAL SUBROUTINES:

FBFIR: Subroutine to check if DUPKEY is legal with fill blanks option.
On exit, if, Q = 0, DUP is valid. Otherwise Q = $4 (error code).

READ1:   Reads the previous record from disk. On entry ATEMP + 3 = 0 means read the remainder only; ATEMP + 3 ≠ 0 indicates read of the whole record. On exit, ATEMP + 2 contains the pointer to the record read, and MODE, is stored in ATEMP + 1.

SUBR: Prepares parameters for the routine DSKUP and then displays.

LOOPY: Extracts the required character: if at left, move the character to the right; if at right, add one to the address. The character is placed in MODE.

INTERFACE

1. ENTRY METHOD

    DUPKEY is loaded via ENTPRG when the operator hits the DUPKEY. Execution begins at the only entry point DUPKEY.

2. INPUT PARAMETERS OR CONDITIONS

    I = Current ATT base address

    SW2 B4 = 1 if BSC or BSF were entered during the current record.

3. EXIT METHOD

    JMP-   (ERROR) when error occurs.
    JMP+   END of 1 when terminal in verify mode.
    JMP+   EMMI when terminal in entry mode.

4. OUTPUT PARAMETERS

    Field duplicated or verified or error message generated.

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME___DUPKEY_____ Level 06

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ZERO ($22)
        ZROBIT ($33)
        NZERO ($12)
        ONEBIT ($23)
        SST ($47)

    b. SST

        CFFA (12)
        STEMP (20)

    c. ATT

        CURCH (2)
        MODE (2)
        FCNT (3)
        RCNT (4)
        MISCH (5)
        TRB (6)
        ATRB (6)
        PFBA (7)
        CURDA (9)
        DT (13)
        BMODE (15)
        SW1 (19)
        SW2 (20)
        FUNDIX (21)
        LUN (35)
        ATEMP (73)

    d. TRANSFER VECTOR

        ERROR ($96)
        DISPLAY ($A0)
        EXDFUN ($9F)
        CAM ($AE)
        WAIT ($B0)
        REDDWA ($A9)
        ENTSR ($91)
        GBVTE ($94)
        SBYTE ($95)

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME___DUPKEY_____ Level 06

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

EMMI	(EMM)	Validate current field and continue processing.
ENDOF1	(VMM)	Check for revalidation and continue verify processing.
BORENT		Set up for new record.
DISPLAY		Handle display output.
DSKUP	(MISKEY)	Display manual Skip/DUP.
DCOL	(HDRNOS)	Display column number.
DUPIT	(SKPDUP)	Adjust ATT counters to point to end of current field.
SETMOV	(ENEXSR)	Clear the program header I/O bit.
WAIT	(ENEXSR)	Delay program.
ENTSR	(ENEXSR)	Common subroutine entry.
EXTSR	(ENEXSR)	Common subroutine exit.
GBYTE	(GSE)	Get character from TRB.
SBYTE	(GSE)	Store character in TRB.
ERROR	(GSE)	Display error at terminal.
EXDFUN	(ENTPRG)	Exit Disk resident function.
CAM		Allocate core Buffer.
REDDWA	(PIO)	Read Disk word addressing.

4. DATA AREAS USED

    ATT, TRB, SST

DOCUMENT CLASS _____CYBERDATA_____

PROGRAM NAME_____DUPKEY_____Level 06

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $182_{16} = 386_{10}$ program length.

   Buffer according to record length or record remainder.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS ___DZR_____

PROGRAM NAME_____Double Zero Key Processor_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    DZR processes the double zero key. In entry or pseudo entry, the format
    is checked to ensure that space remains in the field for two characters.
    If so, DZR stores one via SBYTE and exits to EMM to enter the second zero.
    In verify or pseudo verify, DZR checks the next TRB character for zero.
    If it is not zero, a mismatch error is generated. If it is zero, DZR
    updates the record and field counts and calls VMM to verify the second zero.

2. DETAILED DESCRIPTION

    DZR is a disk resident module called into core by ENTPRG. First DZR calls
    ENEXSR (SETMOV) to clear the I/O bit in the program header to allow the
    program to be moved within allocatable core, if necessary. If this key
    is the first entry in the record (ATT/SW1/2), BORENT is called to perform
    beginning of record initialization functions. A check of field data type
    is made to see that it is not pure alpha. If so, an invalid function key
    error (4) is generated via ERROR. Next a check is made to ensure that
    space remains in the field for two more characters. If not, the same in-
    valid function key error is generated.

    If so, mode checks are made. If the terminal is in pure read mode, the same
    invalid function key error is generated. If the terminal is in entry or
    pseudo entry mode, the first zero is stored in the TRB via SBYTE, the record
    count and field count pointers are updated, the first zero is displayed via
    DISPLY, and control is given to EMM to process the second zero. If the
    terminal is in verify or pseudo verify mode, ATT/SW1/9 (ignore first skip
    flag) is cleared, the next character is retrieved from the TRB via GBYTE
    and a compare for zero is made. If the TRB character was not zero, a mis-
    match character error (5) is generated via ERROR and the terminal mode is
    changed to IDK (ATT/CMODE/14). Otherwise, the record count and field count
    pointers are updated, the zero is displayed via DISPLY, and control is
    given to VMM to verify the second zero.

INTERFACE

1. ENTRY METHOD

    Loaded into core via ENTPRG. Execution begins at the only entry point DZR.

2. INPUT PARAMETERS OR CONDITIONS

    Operator depressed double zero key.
    I = current ATT base address

/5

3. EXIT METHOD

    a. Jump to EMM if legal entry in entry or pseudo entry mode.

    b. Jump to VMM if legal entry in verify or pseudo verify mode.

    c. Jump to ERROR if in pure read mode, no room for two more characters
        in field, field type is pure alpha, or mismatch error.

4. OUTPUT PARAMETERS OR CONDITIONS

    a. Zero displayed to operator and entered in TRB with pointers updated.

    b. Zero displayed to operator and TRB pointers updated.

    c. Error 4 (invalid function key) or Error 5 (mismatch character) errors
        displayed. For error 5, terminal mode changed to IDK.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        GBYTE ($94)
        SBYTE ($95)
        ERROR ($96)
        EXDFUN ($9F)
        DISPLY ($A0)

    b. ATT

        CURCH (2)
        CMODE (2)
        FCNT (3)
        RCNT (4)
        MISCH (5)
        FDTP (13)
        SW1 (19)

    c. SST

        CFFA (12)

2. MACROS USED

    None

16

DOCUMENT CLASS ___DZR___

PROGRAM NAME___Double Zero Key Processor___

3. PROGRAM DEPENDENCIES

ENEXSR	(SETMOV)	Clear I/O bit in header
BORENT		Perform beginning of record initialization
DISPLY	(DCAROL)	Display character and column number Routine which contains all display entry points such as DCAROL
EMM		Perform entry function for the second zero
VMM		Perform verify function for the second zero
GSE	(GBYTE) (SBYTE) (ERROR)	Get the next byte (character) from the TRB Store the next byte (character) in the TRB Display error code and cleanup of TIQ and ATT
ENTPRG	(EXDFUN)	Common disk resident routine exit cleanup routine

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$47_{16} = 71_{10}$ words.

2. RELOCATION TYPE

Run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

/7

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME ___EBCDIC - Conversion routine between EBCDIC/ASCII___   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   EBCDIC converts a string of characters between ASCII and EBCDIC or reverse.
   It is called with a return jump.

2. DETAILED DESCRIPTION

   a. (CNVRT). Save direction of conversion in SST/STEMP + 3.

       1) Turn off I/O-bit to allow module to be moved.

       2) Set STEMP = buffer address of character string.

       3) Set STEMP + 1 = number of words to translate.

       4) Update return address in CNVRT.

   b. (BACK). Get buffer length and subtract one to see if more has to
       be translated. If buffer length is exhausted go to step c.

       1) Translate left half of the word to which STEMP points with
           subroutine ASEBC.

       2) Save translation of left half word in STEMP + 2.

       3) Translate right half of the word to which STEMP points with
           subroutine ASEBC.

       4) Merge the two half-bytes and store the translated word back indirect
           through STEMP.

       5) Update bufferpointer (STEMP) by one.

       6) Go to step b.

   c. (DONE). Restore direction of conversion (STEMP + 3) into A-register.

       1) Decrease user count by one.

       2) Jump back to caller through CNVRT.

   SUBROUTINE ASEBC

   d. (ASEBA). Mask off right half-byte.

       1) If SST/STEMP + 3 $\geq$ 0 (ASCII to EBCDIC conversion) go to step i.

       2) If EBCDIC character $\geq$ $40 (valid character) go to step e.

       3) If EBCDIC character is not the EBCDIC-field separator go to step g.

       4) Set (A) = ASCSEP (code for an ASCII-field separator set to $1C)
           and go to step h.

e. (NOTSEP). If EBCDIC character ≥ 80 go to step f.

   1) Set (A) = ASCII character from TABEC corresponding to the
       EBCDIC character and go to step 4.

f. (NEXT). If EBCDIC character is not in range $\$80 \leq char \leq \$C0$ go to step g.

   1) Set (A) = corresponding ASCII character from TABEBC and go to step h.

g. (SPECIL). Set (A) = $00 (special character code for a non-EBCDIC
    character).

h. (RETB). Mask off higher order byte and exit back to caller.

i. (ASCII). This position of the subroutine performs ASCII to EBCDIC
    conversion.

   1) If ASCII character $< \$IC$ (illegal code) go to step i3.

   2) If ASCII character $< \$60$ (legal code) go to step 1).

   3) If ASCII character = $7C (illegal code) go to step j.

   4) If ASCII character = $7B (pos zero) set (A) = $C0 and exit out
       through step k.

   5) If ASCII character = $7D (neg zero) set (A) = $D0 and exit out
       through step k.

   6) If ASCII character = anything else go to step j.

j. (SPE). For code for a nonexisting ASCII code.

k. (RETA). Go to step h.

l. (TABLE). Get word count into conversion table TAB and access table
    accordingly. If character count is even shift character to Cover
    Byte go to step h.

INTERFACE

1. ENTRY METHOD

    Loaded into core by ENTPRG. Is started by means of a return jump to
    entry point CNVRT.

2. INPUT PARAMETERS OR CONDITIONS

    RTJ- $,Q
    ADC   buffer address of string to convert
    ADC   number of words to translate
    Where   (A) positive for ASCII-EBCDIC conversion
                negative for EBCDIC-ASCII conversion
            (Q) address of last word in header.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME_____ __EBCDIC - Conversion routine between EBCDIC/ASCII__   Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $B5_{16} = 181_{10}$ words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

ANALYTICAL SUPPLEMENTS

Conversion tables.

ASCII TO EBCDIC

TAB	NUM	$2200	ASCII	1C-1D	FS-
	NUM	$0000		1E-1E	
	NUM	$405A		20-21	SPACE - 1
	NUM	$7F7B		22-23	' - #
	NUM	$5B6C		24-25	$ - %
	NUM	$507D		26-27	% - '
	NUM	$4D5D		28-29	(-)
	NUM	$5C4E		2A-2B	* +
	NUM	$6B60		2C-2D	, -
	NUM	$4B61		2E-2F	. /
	NUM	$F0F1		30-31	0 1
	NUM	$F2F3		32-33	2 3
	NUM	$F4F5		34-35	4 5
	NUM	$F6F7		36-37	6 7
	NUM	$F8F9		38-39	8 9
	NUM	$7A5E		3A-3B	: ;
	NUM	$4C7E		3C-3D	< =
	NUM	$6E6F		3E-3F	\ ?
	NUM	$7CC1		40-41	@ A
	NUM	$C2C3		42-43	B C
	NUM	$C4C5		44-45	D E
	NUM	$C6C7		46-47	F G
	NUM	$C8C9		48-49	H I
	NUM	$D1D2		4A-4B	J K
	NUM	$D3D4		4C-4D	L M
	NUM	$D5D6		4E-4F	N O
	NUM	$D7D8		50-51	P Q
	NUM	$D9E2		52-53	R S
	NUM	$E3E4		54-55	T U
	NUM	$E5E6		56-57	V W

ASCII TO EBCDIC (CONTD)

NUM	$E7E8	58-59	X Y
NUM	$E9C0	5A-5B	Z
NUM	$E0D0	5c65D	
NUM	$5F6D	5E-5F	-

EBCDIC TO ASCII

NUM	$205B	EBCDIC	40-C0	SPACE 0+
NUM	$0041		41-C1	- A
NUM	$0042		42-C2	- B
NUM	$0043		43-C3	- C
NUM	$0044		44-C4	- D
NUM	$0045		45-C5	- E
NUM	$0046		.46-C6	- F
NUM	$0047		47-C7	- G
NUM	$0048		48-C8	- H
NUM	$0049		49-C9	- I
NUM	$0000		4A-CA	-
NUM	$2F00		4B-CB	. -
NUM	$3C00		4C-CC	< -
NUM	$2800		4D-CD	(-
NUM	$2B00		4E-CE	+ -
NUM	$0000		4F-CF	
NUM	$265D		50-D0	%- 0-
NUM	$004A		51-D1	- J
NUM	$004B		52-D2	- K
NUM	$004C		53-D3	- L
NUM	$004D		54-D4	- M
NUM	$004E		55-D5	- N
NUM	$004F		56-D6	- 0
NUM	$0050		57-D7	- P
NUM	$0051		58-D8	- Q
NUM	$0052		59-D9	- R
NUM	$2100		5A-DA	] -
NUM	$2400		5B-DB	$ -
NUM	$2A00		5C-DC	* -
NUM	$2900		5D-DD	) -
NUM	$3800		5E-DE	; -
NUM	$5F00		5F-DF	
NUM	$2D5C		60-E0	- <
NUM	$2F00		61-E1	/ -
NUM	$0053		62-E2	- S

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __EBCDIC - Conversion routine between EBCDIC/ASCII__   Level 6

EBCDIC TO ASCII (CONTD)

NUM	$0054	63-E3	- T
NUM	$0055	64-E4	- U
NUM	$0056	65-E5	- V
NUM	$0057	66-E6	- W
NUM	$0058	67-E7	- X
NUM	$0059	68-E8	- Y
NUM	$005A	69-E9	- Z
NUM	$0000	6A-EA	
NUM	$2C00	6B-EB	, -
NUM	$2500	6C-EC	% -
NUM	$5F00	6D-ED	-
NUM	$3F00	6E-EE	\ -
NUM	$3F00	6F-FF	? -
NUM	$0030	70-F0	- 0
NUM	$0031	71-F1	- 1
NUM	$0032	72-F2	- 2
NUM	$0033	73-F3	- 3
NUM	$0034	74-F4	- 4
NUM	$0035	75-F5	- 5
NUM	$0036	76-F6	- 6
NUM	$0037	77-F7	- 7
NUM	$0038	78-F8	- 8
NUM	$0039	79-F9	- 9
NUM	$3A00	7A-FA	: -
NUM	$2300	7B-FB	# -
NUM	$4000	7C-FC	@ -
NUM	$2700	7D-FD	: -
NUM	$3D00	7E-FE	= -
NUM	$2200	7F-FF	" -

AA5641   PRINTED IN THE USA

3. EXIT METHOD

   Exit back to calling routine.

4. OUTPUT PARAMETERS OR CONDITIONS

   The contents of the specified buffer is converted as specified. A-register
   is conserved.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK ($2)
      NZERO ((12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)
      SST ($47)

   b. SST

      STEMP (20)

   c. Special codes

      ASCSEP ($1C)   ASCII field separator
      EBCSEP ($22)   EBCDIC field separator
      ASCSPE ($00)   Code for a non-EBCDIC character
      EBCSPE ($00)   Code for a non-ASCII character

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   None.

4. DATA AREAS USED

   SST
   Buffer area as specified in calling sequence.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ECD - Enter Check Digit___   ___Level 6___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   ECD is entered by the supervisor (at the console) in order to establish one
   check-digit test table. The modulus and weights to be applied are read
   and the CWT (CDV Weight Table) is built according to the corresponding
   values, keyed in by the supervisor. If less than 15 weights are specified,
   weights of 0 are given to the high-order digits of the field.

2. DETAILED DESCRIPTION

   ECD starts by calling the GETALP subroutine and checking that "ECD"
   command is followed by a comma. If not, an error message is sent to
   the console and the command is rejected. Next, the first number (Test
   Number) is read and checked for validity. A nonnumeric or too great a
   value results in an error message sent to the console, as well as a zero
   value or a missing one. The Test Number should not exceed the maximum
   CDV Test Number stored in word 3 of the EST. If the value supplied was
   correct, ECD proceeds by setting a pointer (SAVE) to the correct
   displacement in the CWT, corresponding to the given Test Number (each
   test is assigned a 4-word storage area). That storage is cleared and
   a loop is entered during which successive numbers are read and stored
   in their corresponding place in the actual CWT. The first value after
   the Test Number is the modulus and a check is made for correctness (the
   modulus must be less than 16 and nonzero).

   During the loop, each number read from the input unit is added to a
   temporary word (WØRD) which is then shifted 4 digits left. After each
   4-values, WØRD is stored in the table, cleared, and the pointer SAVE
   is increased by 1. Upon detection of an error (values greater than 16
   or nonnumeric) a corresponding error message is sent to the console and
   the command is rejected through REJSUP. Before returning control to the
   SV routine, the resulting CWT is written onto disk, according to the
   addresses specified in the EST.

   Register I is updated to point to the ATT base address.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___ECD - Enter Check Digit___   Level 6

INTERFACE

1. ENTRY METHOD

   From DSKSUP, through ENTPRG

2. INPUT PARAMETERS OR CONDITIONS

   I = current ATT base address.

3. EXIT METHOD

   If no errors (in command syntax, test number, modulo or weights), control is returned to SV. Missing weights are set to 0.

   If any error was detected, a corresponding message is sent and control returns to REJSUP.

4. OUTPUT PARAMETERS

   Updated CWT written on disk.

   I = current ATT base address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK ($2)
      ZERO ($22)
      NZERO ($12)
      ONEBIT ($23)
      SV ($B1)
      ATTB ($10)
      SUPRW ($97)
      GETBIN ($99)
      SST ($47)
      REJSUP ($B2)
      WRITSEC ($A8)
      GETALF ($9B)
      GETCOM ($9C)

b. ATT

FW (3)
ACWT (2)
CWTDA (21)
SAVE (46)
WORD (48)
FLAG (50)
MAXCDV (3)
STEMP (20)
SUPCOM (36)
CNT (47)
CFOUR (49)

C. CONSTANTS

None.

2. MACROS USED

None.

3. PROGRAM DEPENDENCIES

GETALP - entry in SCANV to retrieve commas
REJSUP - entry in SUPER used when error exit is taken
GETBIN - entry in SCANV to retrieve binary numbers
WRTSEC - entry in PIO to write the new CWT table
SV   - entry in SUPER for normal exit

4. DATA AREAS USED

ATT
SST
EST

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$07C_{16} = 124_{10}$ words.

2. RELOCATION TYPE

Loader relocatable, run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___EDC - Enter Document_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    EDC processes the supervisor command EDC - enter a document specification.

    Command syntax:

    $$EDC,nnn(1),Fnnnn(2),R \begin{Bmatrix} [nnnnn(3)] \\ unspecified \end{Bmatrix} \quad [,Fnnnn,R \begin{Bmatrix} [nnnnn] \\ unspecified \end{Bmatrix} \quad .....] \quad CR$$

    where:

    (1) document number (1 through 255).

    (2) format number of first record type.

    (3) number of times this format appears.
        Fnnnn and Rnnnnn are repeated until the last format has been
        specified. If the parameter string is longer than one line, a
        comma and CR are entered immediately after the R[nnn]. When
        an unlimited number of records are specified for a format, no
        number is entered after the 'R'.

2. DETAILED DESCRIPTION

    EDC is a locked function. On calling it, its lock is checked and a
    wait loop is started, if the program is busy; else the lock is set and
    processing starts. EDC starts scanning the command parameters, checking
    every parameter read for syntax errors. The document number is fetched
    and the entry for that document is read from the LDD, to ensure that no
    other document has that number. From the header of the LDD, the next
    available word in the LDT, and the maximum size are found. Before trying
    to add an entry to the LDT, EDC makes sure that there is space left in
    the table. CAM is called for allocation of a sector size buffer. The
    details to be added to the LDT are located in this buffer, and whenever
    it is full, or end of parameters list is detected, WRTDWA is called for
    writing the buffer in the table. Whenever a format number is scanned,
    it is checked against the maximum format number allowed in the system,
    and the entry for that format is read from the LFD. It must be an
    existing format.

    If just after the record number (or after the R), a CR is entered, it
    signals EDC, that the end of the parameter list has been reached. In
    that case all the parameters that are still in the buffer are written
    into the LDT with a sign on bit 15 of the last word written.

The address of the next word available in the LDT is stored in the first
word of LDD, and the entry for that document in the LDD is activated.
Now the buffer is released, lock reset and control is passed to SV. If
the parameter list occupies more than one line (where a CR is entered
after R[nnn],), EDC checks whether the input unit is a CRT on the TTY. If
it is the TTY, a line feed is printed, and a new line is read into the
Interrogate buffer. If it is a CRT, DISPLAY is called to move the line
which has previously been entered, to the top of the screen.

Reading and writing to the CRT terminal is done through SUPRW (an entry
point in PIO).

Reading from disk (LDD and LDT), is done through REDDWA (in PIO).
Writing to disk is done through WRTDWA (in PIO).
When an error is detected, control is passed to REJSUP with the error
code in the A register.

INTERFACE

1. ENTRY METHOD

    EDC is a supervisor function processor loaded by ENTPRG. Execution
    begins at its only entry point EDC.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.
    Other parameters are entered by the operator, according to function syntax.

3. EXIT METHOD

    If no error was detected during processing exit is made to SV. Else
    jump to REJSUP with error code in A.

4. OUTPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

    Error messages:

    a. INVALID PARAMETER

    b. INVALID FORMAT NUMBER

        Format no. greater than the maximum allowed, or format no. was not
        activated.

    c. INVALID COMMAND FORMAT

d. MEMORY FULL-RETRY

CAM did not succeed in allocating a buffer.

e. INVALID DOCUMENT NO.

A nonnumeric field for doc. number, or doc. number greater than 255, or doc. number already active.

f. DOCUMENT DIRECTORY FULL

g. I/O ERROR

When reading a line.

h. INVALID NO. OF RECORDS

i. FUNCTION TERMINATED

Cancel bit was on.

j. FUNCTION BUSY

ASSEMBLY CHARACTERS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ONEBIT ($23)
        GETALP ($9B)
        GETBIN ($99)
        GETCOM ($9C)
        REJSUP ($B2)
        SUPRW ($97)
        SV ($B1)
        WAIT ($B0)
        ENTSR ($91)
        EXTSR ($92)
        DISPLAY ($A0)
        CLCDWA ($90)
        CAM ($AE)
        REDDWA ($A9)
        WRTDWA ($AA)
        ZERO ($22)
        SST ($47)

    b. SST

        STEMP (20)

    c. ATT

        DOCNR (48)
        POINTER (49)
        SECIX (51)
        TABLIM (52)
        FMTLIM (53)
        LENG (54)
        TEMP1 (55)
        TEMP2 (56)
        WORD1 (46)
        WORD2 (47)
        LDTENT (58)
        CORADD (57)
        ATEMP (73)
        AINT (27)
        TERMIN (23)

    d. EST

        LDDDA (39)
        MAXFMT (24)

    e. CONSTANTS

        LU (68)
        FR (2)
        FW (3)

2. PROGRAM DEPENDENCIES

SCANV	GETBIN:	To retrieve numeric parameters
	GETALP:	To retrieve alpha parameters
	GETCOM:	To check for commas
PIO	CLCDWA:	Calculates disk word addresses
	REDDWA:	Read from disk using word addresses
	WRTDWA:	Write to disk using word addresses
	SUPRW:	Output to CRT
SUPER	SV:	Normal exit
	REJSUP:	Error exit
ENEXSR	WAIT:	Wait for function release
	ENTSR:	Store return address in stack
	EXTSR:	Return via stack
DISPLAY	SETDIS:	Set display indicator
CAM		Obtain memory space
SETMOV		Enable program move

AA5641   PRINTED IN THE USA

3. DATA AREAS USED

    ATT
    SST
    LDD
    LDT
    EST
    LFD
    INTERROGATE BUFFER
    CAM

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    Program = $(352)_{10}$

    CAM buffer = $(99)_{10}$

2. RELOCATION TYPE

    Run anywhere

3. ENTRANCY CLASSIFICATION

    Locked, not reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS _____ CYBERDATA

PROGRAM NAME_____ EFM - Enter Format _____ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

EFM is entered by the supervisor when a new format has to be specified
or an old one has to be revised. The program sets up the corresponding
format tables, according to the parameters passed by the supervisor, and
writes them on disk.

2. DETAILED DESCRIPTION

Follows the program listing, because of program's complexity.

EFM consists of 4 major parts:

a. Monitor - this part performs specification records reading, checks
for I/O errors and calls the FMT processing subroutine.

b. FMTENT - specification records processing. Calls analysis routine
to analyze records. After the last specification record,
overall checks are performed to ensure format consistency.
If no errors are detected, the format is written on disk.

c. ANALYS - analyzes a single record. Interprets the control table
information as pseudo instructions, and takes appropriate
actions.

d. CONTAB - calls numeric or character-processing subroutines or bit-
setting macros, according to the word to be interpreted.

EFM starts by making itself movable, setting a flag (HEADER bit 15) to
indicate primary read and clearing both the format string base address
and the FBA pointer (ATT word 7). ATT word 68 is checked for any input
errors (first line read by SV).

AA5641

Description Following the Program Labels

READ1:   Increase the current field number. Clear bit 0 of ALLAUT
            (no auto flag).

READ1L: If the supervisor input unit is a CRT, its status is set to
            interrogate mode and 24 words are read. For a TTY unit, a line-
            feed control character is sent and 40 words are read. If the
            cancel key was entered, the previous situation is restored
            (GØERO1 with A=9) and a new line is read (READ1). Otherwise,

CKIO: If an input error occurred, a message is sent (through ERROR with
            error index = 9). Otherwise,

READ1A: If CRT control goes to PRØCES. If TTY, the actual number of
            characters read is stored in word 30f the interrogate buffer
            (80 if no short read), and then

PRØCES: If bit 13 of HEADER is set (secondary read) control goes to READL.
            If primary read, the FMTENT routine is called.

ENTRTN: If no error in last line and A is no 0, control goes to READ1
            (read in a new line). If A is (-); an error occurred in the
            processed line, control passes to ERRØR. If A = 0 then

EXIT: Release the requested area now used by the actual format, by
            setting its length in word 2 of that space. Clear CALRPI in CFM
            (caller's program index), release the area occupied by EFM
            (through EXDFUN) and write the format on disk (through ENTPRG).

ERRØR: If supervisor list unit is a CRT, check if end of screen has
            been reached: if yes, clear the screen. If the error code
            passed is not zero, control goes to ERØR3. Else, store the
            erroneous character (the last one received) and the column
            number in the message 'Character x illegal for parameter yy'.
            That message is moved to the interrogate buffer by a loop named
            ERØR2 and then it is printed. Pass control to ERRØR 5.

ERØR3:   Select an error index from a given table (ERIDX) according to
            the given error number (0 to 22) and print the corresponding
            message. Then,

ERØR5:   If that line was the header or the error number passed was 9,
            release the present format space (through RELFØR) and return
            to REJSUP, Else,

ERØR8:   If at least 4 parameters were entered (COLUMN   4) then the
            record counter (RECCNT) is decreased by the maximum field
            length. And in any case,

ERØR8A: All the space in the FMT required by that field is cleared, the
            field number is decreased by 1 and the format pointer (FMTPNT)
            set to the third word in the current field table, after which
            a new input line is read (READ1).

FMTENT: (the format sub monitor)
            The parameter number is set to 0, HEADER bits 0 - 14 cleared and
            bit 15 is set, control passed to FMTMD (Format Header).

FMTLIN: The input buffer pointer (INPTR) is set to 0 and the ANALYS routine
            is entered, with A set to the displacement between FLDINF and
            LØDCNT.

SETLNG: Current field length is calculated (FMTPNT - FCDBAS) and after
            adjustment and insertion of previous field length it is stored
            in the field's first word. Current length replaces the previous
            one (shifted 4 bits left).

CKØV: If FMTPNT (current word in format table) exceeds 510, an error
            code of 2 is passed to GØER through A. Otherwise,

NØØV1:   If current field is not an auto-field (bit 0 of ALLAUT=1) then bit
            14 of ALLAUT is set. In both cases

MØØV1:   A check is made for the verification type (field word 1), when it is
            not 0, FMT word 3 bit 14 is set to indicate verification is needed,

CHLF: If that was not the last field (RECCNT less than or equals to
            record length) a new line is read (through ENTBCK, with A positive).
            Otherwise,

LSTFLD: Number of words in current field entry is cleared. If bit 14
            of ALLAUT is 0, the format is rejected (through GØER). If not,
            RESEQ is checked. When it contains a positive value, ENF exits
            (through ENTRTN with A=0). When bit 15 of RESEQ is set
            (resequencing needed) then

CKRSQ1:   If autosequencing exists in the format (word 1 bit 13 set), current
            field number is increased by 1 (last field indicator) and NØFLD
            set to one. Two loops are entered. The inner one (CKRSQ3) looks
            through all field entries in FMT for a resequence field number
            corresponding to MØFLD. The outer loop (CKRSQ2) increases NØFLD and
            starts the searching again, until either all field numbers have been
            found (NØFLD = FIELD), which signals that resequencing is correct,
            or the last field entry has been reached without finding NØFLD
            (that is, some field number is missing), and then the command is
            rejected (ENTBCK with A negative).

FMTMD: Processing the header line.
Temporary variables are cleared. If there is no comma after
'EFM' the command is rejected (invalid command format).

FSIZE: CAM is called to achieve core for maximum format table size. If
not available, another try is made each Cyberdata cycle, up to
100 times. If core is still not available, EFM gives up and
sends an error message - MEMORY FULL - RETRY. If CAM returns the
address of available space, it is linked back by saving the FMTBAS
address on top of the FMT (as backwards pointer) and FMTBAS points
to top of FMT.

CLRLØP: The achieved core area is cleared (554 words), not changing the
first three words in FMT and the ANALYS routine is entered with
A set to the displacement from LØDCNT to MDRINF.

RTNHDR: Set the third word of FMT to correct values (according to the
information contained in:

              FREVER - forces reverification
              FMTNØ - format number) and mark it as active (bit 15 = 0).
Set maximum record length (minus 1) in word 2 of FMT. Set
autosequencing flag bit (#13) and clear time control bit in FMT
word 1, according to AUTSEQ. If no autosequencing required,
header and all-auto flags are cleared and control passes to
either SETLNG (set auto sequence length) or GETNXT (get first
field) depending on whether Q is zero or not. If autosequencing
is required then,

SETBAS: Set field length to 3 (field format word 0), set data type to pure
numeric and keying mode to variable. Also set field name length
to 3 words. If resequencing is required, the corresponding
resequence number (lower half of RESEQ) goes to field table
word 3, the RS bit (word 2 bit 15) is set and Q is increased by 1
(FMT word 7 is already used). It assures that in both cases
(autosequencing or not),

NØRS: Places the field name AUTSEQ in its right place, before transferring
control to SETLNG.

ANALYS: The input line analyzing routine. Control table relative address is
stored in CMTPTR. If supervisor list unit is a CRT and that was
last line, the screen is cleaned. The input line is rewritten
except for 3 cases: supervisor list unit is also its input unit,
number of characters read is 0 or the line was a header.

ANLY: A pointer is set to the input buffer parameter number (4 if header line, 0 if not). One character is read in and if it is not on ASCII character (end-of-record), default values are set or not depending on the buffer containing at least 3 parameters before the end-of-record is found. A loop (between UP1 and PCHAR) counts the number of parameters entered. If the 'REVISE' bit is set, jump to REVISE even when no parameters were entered before EOR.

EØR: When a space (not the fill character) is detected, a test is made to find the end of record after it. If not found, program continues counting the parameters and a later check will find that error.

SET: If more than 16 parameters were keyed in, control passes to END (no need for default parameters). If some parameters were omitted, a loop (NEXT) is entered during which the remaining parameters are transferred from a default-parameters table (TABMDR) to the input line (the interrogate buffer), through the STØRE subroutine. When the resequencing default parameter is met, then,

RESEQN: The current field number (+1 if autosequencing specified) is translated to BCD (by the external BD routine), then to ASCII characters and is stored in the input line. Control returns to NEXT.

STØRE: Stores in the input buffer (in the right halfword) the character passed in A. When end-of-record is detected, control is passed to

ANALYSE: The parameter number is increased and a control word is loaded from the control table (MDRINF for header, FLDINF for field line). If (according to the control word) the parameter has to be numeric, the GETSUB subroutine is called and the ANLYSE loop is entered again. If an alphabetic field then,

ALPHA: A loop is entered during which the several possibilities defined for the actual parameter are checked against the corresponding parameter in the input buffer. A flag (MATCH) is set to 1 and cleared as soon as a correspondence is found. Then either the SETBIT subroutine is called (for setting a bit in the FMT) or a numeric subroutine is called (by GETSUB) to process the numeric information in that parameter. Errors are signaled when:

- the input buffer contains an erroneous parameter or an illegal ASCII character;

- no match is found with any possible parameter.

If the parameter ends with a comma, control returns to ANLYSE and the next parameter is checked. If not,

ALPHA8:   The next character is retrieved. One or more spaces followed
            by end-of-record (or the EØR alone) pass control to ALPHA9.
            Otherwise, an error is signalled (through GØER1).

ALPHA9:   If header line, EØRHDR will resume processing it. If not,

EØRFLD: If all 16 parameters have not been specified or the lower and
            upper limits were not given, ERR7 signals an error. If the parameters
            were correct,

EXANYL: In the case of header processing, RTNHDR receives control.
            Otherwise the field name (if specified) is transferred to the
            FMI, and control is passed to RTNFCD.

EØRHDR: If the autosequencing parameter is not given or it was given and
            resequencing needed but the autosequencing field number was not,
            an Incomplete Command error message is issued. Otherwise,
            control passes to EXANYL.

SETBIT: If bit 15 of the parameter following the call is set, SETBIT
            exits without any other action (just skipping the parameter).

            If not, the corresponding bit (B) is selected and inserted into
            the word (W) of the actual field in FMT. B and W are parameters
            of the TSTB macro.

GETSUB: After saving the returning address (for possible future I/Ø through
            ENTSR, the actual subroutine address is retrieved from an
            addresses table (SUBTAB) and control is passed to it. Return is
            via EXTSR.

REVISE: This section deals with format revision. Extract field number and
            format string length. Subtract format field length from string
            length, and look for field requested, adjusting for autosequencing.
            If not found, exit to error (NO FIELD TO COPY). Else check for
            Resequencing error and Record overflow which cause error exits.
            If no errors are detected, move field information to new format,
            and clear previous and current format field lengths. Jump to
            SETLNG.

SUBTAB: Table of relative addresses to the specified subroutines.

SUBROUTINES:

S00: Set bit 15 of FMTNØ (revised format).

S01: Get format number (through GETNUM). If zero, signal an error. If not zero, insert it in FMTNØ, and compare to maximum format number allowed. If greater, signal an error. Else enable EFM to move (ENTSR), check if format exists (GFCØRE). An error is signalled if format already exists (new format) or when trying to revise a nonexistent one.

S02: Read maximum record length (GETNUM). If zero, signal error 13 (Illegal record count). If revised format, the actual maximum length has to be equal to the one kept in RECMAX. Otherwise, an error is sent. If they are equal, the corresponding format is released (by RELFØR). If new or revised format,

S022: Compare to the system's maximum record length (MAXRC). If lower or equal to it, the actual record length is correct. Otherwise, an error message is issued.

S03: Set the FREVER (Forced reverification-bit number 13).

S04: Set bit 15 of RESEQ (resequencing needed).

S05: Set the autosequencing flag (bit 14 of AUTSEQ). The maximum record length has to be at least 5 (and then RECCNT starts from 4), otherwise an error is sent.

S06: If resequening or autosequencing are not required, return with no other action. Autosequencing field number zero or greater than 127 is illegal and results in an error. Else, the resequence number is added to ATEMP.

S07: Exit from analyzer (retrieve return address from stack).

S08: Get field name. Blank ATEMP to ATEMP+3.

S082: A loop during which up to 6 characters are read and stored in their correct halfword (ATEMP+1 to ATEMP+3). A name longer than 6 characters is erroneous.

S087: The last right halfword (if not used) is blanked.

S088: If name ends with blanks, reduce its length so it will show only meaningful words in FMT; then store the length in word 2 of actual field in FMT.

GETNUM: A number is read. If nonnumeric, an error message is sent. If autosequencing field number and followed by comma, error 21 is sent. If not last parameter and not followed by comma an 'Incomplete line' error message is sent. Else return to caller.

S09: Read a number (resequencing field number). If resequencing is not requested, return to caller. If number greater than 127, error message 4 is issued. Else set the RS bit in current field table and store that number in the next free halfword in FMT (done by STRHW). Then return to caller.

SOA: Read maximum number of characters. If zero, or greater than 100, issue error message 15. Else store it in FMT. Get data type. If signed numeric (111) and length is 0, error 0 is sent.

SOA21: Increase RECCNT by length of that field. If now RECCNT exceeds RECMAX (specified in format header) error 5 is signalled. Else return to caller.

SOB: Increase ALLAUT by 1 (current field not an auto one).

SOC: Get data type (by SOC1). Return to caller if:

● Last character is R and data is pure numeric or

● Data is alphanumeric and last character is not R (left blank fill wanted). In other cases, send an error message.

SOC1: Get data type (bits 13;15 of word 1 in current field table).

SOD: Get keying mode (bit KM of FMT). If it is 1 (variable), and both boundary check is 0 (not needed) and verification type is 0 (no verification) then return to caller (field can be auto-skipped). Otherwise an error results.

S19: Get data type via SOC1, and check for unassigned numeric, and variable keying. If one of these conditions is not met, exit to error.

SOE: Get one character. Restore comma if necessary (when A = 0). If that character is 'N' and is followed by a comma, return to caller. Otherwise, send error message 21. If character is not N but recording mode is variable, an error results (not reasonable to fill blanks to zeroes and then cut them out). Else set the FC bit (word 2 of current field table), store the fill character (in word 3) and check if a comma follows. If yes, return to caller. If no, error.

GETCH: Get one character (through GETALP) and return to caller only if it is a legal ASCII (or comma). Otherwise send an error message.

SOF: Read own code number (through GETNAM). If zero return to caller without any action. If not zero, compare to the maximum own code number (in EST). If lower or equal to it, set bit 13 (ØC) in word 2 of current field table, store the own code number in the next halfword and return to caller. Else send error message 16.

S10: Read special test number (through GETNUM). If zero (no special test wanted) return to caller. If not zero, compare to the maximum special test number (in EST). When higher, send error message 17. If legal, compare it to maximum CDV test number. If higher, set the ST bit (word 2 of current field table), store that number in next halfword in FMT and return to caller. If not higher (CDV test required) then get data type (by SOC1) and see if it is unsigned numeric (data type = 6). If not, send error message 17. If yes, check field length. When greater than 15, error 17 occurs, else save that test number and set the ST bit before returning to caller.

STRHW: Set the bit passed in A, in word 2 of current field table (in FMT). Reverse MLFWRD bit 14 (left or right halfword indicator). When it is 1 store the value in ATEMP into the left halfword of the word FMTPNT points to. Otherwise, store it in the right side of that word and then increase FMTPNT, before returning to caller.

S11: After saving CHAR (passed as parameter), read next character. If comma, return INPTR to point to that comma (as it was '0'). If not comma, check that the counter number is in the allowed range (0 to 7), and signal an error if not. Otherwise A gets the counter number and

S112: See if last function (CHAR) is 'N'. If yes, only counter number 0 is allowed, other counter numbers results in an error. When function is not N, but counter number is 0, error. Otherwise set bit 4 (subtract) only if CHAR is 'S'. In both cases (S02 A) insert the new counter number in CØUNT. If it was 0, return to caller. If not, check (through S1116) if the field has the right length (counter keeps up to 15 digits, so the field cannot be longer). For unsigned numeric fields, maximum length of 15 is allowed, and 16 for signed numeric. Longer or nonnumeric fields cause an error message to be sent.

S18:.   If no counter information specified, return to caller. Otherwise
         set the CNT bit and insert the halfword containing the counters
         information (CØUNT) in the next free halfword in FMT. Then clear
         CØUNT and return to caller.

S15:   If CHAR (now parameter 16) is 'E' then set CØUNT, to 6. Otherwise,
         set it to 4. Check if field is legal (numeric, length less than
         15 or 16) by S1116. If correct, return to caller. If not, send an
         error message.

S16:   Get limits (when specified).

         If CØUNT is 0, an error message is sent (CØUNT should indicate
         inclusive or exclusive operation). If next character is end-of-
         record, the secondary read flag (bit 13 of RETURN) is set and a new
         line is read (through READ 12).

READL: Clear the secondary read flag. If necessary, clear the supervisor
         CRT screen (RDSP2) and redisplay the information (RDSP3) only if
         the supervisor list unit is not the same as its input unit (TTY).
         Restore the return address, altered by the I/Ø (READS1).

S163:   Get the 2 limits (by GETLIM, which stores them in the FMT). After
         restoring pointers (into the FMT: FMTPNT) raise the L (limits)
         bit in FMT and store the number of words for both lower limit and
         upper limit in the next halfword (by STRHW). Then check for
         end-of-record occurrence. If found, return to caller if not,
         send an error message.

GETLIM: Clear a temporary area (SL to SL+3). If end-of-record is the next
         character, send an error message. Else restore the interrogate
         buffer pointer (INPTR) and

GETDIG: · Get one character. If it is a digit (greater or equal $30, lower
         than $34) push it in the rightmost 4 bits of SL, shifting all other
         bits left as if SL to SL+3 was a continuous field of 4 bits
         digits (SL+3 first, SL last, total 16 digits). That task is
         achieved by a series of shifts (GETL1). The leftmost 4 bits of
         SL+3 are lost. If they were not zeros (more than 16 digits specified)
         an error message is sent.

         As soon as a nonnumeric character is read, a test is made at
         GETD2 (it has to be '+' or '-'). If not, an error message is sent.
         Otherwise the appropriate sign is set (F for '-', zeros for '+').
         The last significant word (from SL to SL+3) is retrieved and the
         length adjusted accordingly (GETD4). The sign is inserted (LAST),
         the actual number of words is stored in CØUNT (LAST1) and a loop
         (GETD5) is entered during which the SL temporary area is moved to
         FMT in reverse order (the most significant digits were in the last
         significant word in SL). Then control is returned to caller.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME__ EFM - Enter Format   Level 06

    S17:   No operation subroutine. Does nothing, just returns to caller.

    MDRINF: A table of parameters defining the several possible inputs for every parameter and the required action in each case. The modular table is set up by macros and defines the required inputs and actions for the header line.

    FLDINF: Same as HDRINF, but defines parameters for a field defining line. Refer to the macros descriptions for more information about each one.

INTERFACE

1. ENTRY METHOD

    EFM is entered through ENTPRG, by DSKSUP.

2. INPUT PARAMETERS

    I = current ATT base address
    ATT + 36 = supervisor input unit
    ATT + 37 = supervisor list unit
    ATT + 39 = supervisor basic input unit

3. EXIT METHOD

    If errors are detected in the header line, an appropriate message is issued via REJSUP. If errors occur in field specifications, an error message is issued, but control is retained, unless CNCL or SX is entered, which terminates the function. Otherwise, exit to ENTPRG with an indication for calling 'write format' (write the new or revised format on disk and link it to LFD).

4. OUTPUT PARAMETERS

    a. Error codes are transferred in A.

    b. Updated FMT is written on disk. Current FMT (if revision required and FMT in use) is not altered.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME __EFM - Enter Format__   Level 06

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK ($2)
        ZRØBIT ($33)
        SST ($47)
        NZERØ ($12)
        ZERØ ($22)
        ØNEBIT ($23)
        TEN ($46)

    b. ATT

        STEMP (20)
        ATEMP (73)
        AINT (27)
        SUPINP (36)
        CALRPI (42)
        HLFWRD (47)
        INPTR (48)
        PREVL (51)
        ALLAUT (54)
        AUTSEQ (55)
        FIELD (56)
        CHRCHT (59)
        CØLUMN (62)
        PFBA (07)
        RQ (65)
        TRMNT (23)
        SUPLST (37)
        FMTBAS (46)
        RETURN (47)
        FMTPNT (49)
        RECMAX (52)
        FMTNØ (54)
        FREVER (55)
        CNTPTR (57)
        CHAR (60)

```
        FMTNMB (13)
        LU (3)
        SUPCØM (36)
        BSCINP (39)
        HEADER (47)
        ERCØDE (47)
        FLDBAS (50)
        RECCNT (53)
        RESEQ (55)
        RSQAUT (55)
        MATCH (58)
        CØUNT (61)

    c.  CONSTANTS

        MAXSPT (3)
        MAXFMT (24)
        E ($45)
        B ($42)
        L ($4C)
        S ($53)
        Y ($59)
        RSQASQ (7)
        FW (3)
        MXWAIT (100)
        MAXØC (4)
        A ($41)
        F ($46)
        X ($58)
        N ($4E)
        U ($55)
        LIMTCL (16)
        MCHRCL (3)
        MØT (5)
        MAXRC (38)
        D ($44)
        II ($49)
        K ($4B)
        R ($52)
        V ($56)
        ASQCL (6)
        FR (2)
        SIZE (510)
```

2. MACROS USED

   a. HEAD T,B

      If T = N then execute subroutine SBxx ('B')
      If T = A then B is an index given to each parameter.

   b. ENDF N

      N equals 'B' (second parameter of HEAD). That macro closes the definition
      beginning with HEAD.

   c. TSTB C,B,W,V

      If current character checked is C, then set bit B of word W in
      FMT to the value V.

   d. TSTF C, SBxx

      If current character checked is C then call subroutine Sxx.

   e. SUBR xx

      Set the distance from the calling point (it is fixed) to the subroutine
      Sxx.

3. PROGRAM DEPENDENCIES

   a. EXTERNALS

      SETDIS
      CCKEND
      SETMØV
      BD
      PRTMSG

    b. Transfer Vector

        ENTSR ($91)
        RTNSR ($98)
        GETCØM ($9C)
        EXDFUN ($9F)
        CAM ($AE)
        REJSUP ($B2)
        EXTSR ($92)
        GETBIN ($99)
        GFCØRE ($9D)
        DISPLY ($A0)
        WAIT ($B0)
        SUPRW ($97)
        GETALP ($9B)
        ENTPRG ($9E)
        RELFØR ($AD)
        SV ($B1)

4. DATA AREAS AND TABLES USED

    ATT
    ENTSR Stack
    EST
    Interrogate buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $606_{16} = 1542_{10}$ words.
    528 words for FMT buffer.

2. RELOCATION TYPE

    Loader relocatable, run anywhere.

3. REENTRANCY

    Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____EJB - Enter Job_____Level 6____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

The EJB program processes the EJB supervisor command. Its function is to create a new entry in the LJD, or to modify an existing entry.

2. DETAILED DESCRIPTION

Before calling the GETLJD program, whose function is to create or modify the specified entry in LJD, EJB checks the validity of the given parameters. The parameters are inspected in the same order as they appear in the command. The first erroneous parameter will cause an error exit to be taken to REJSUP. If all parameters are valid EJB calls GETLJD, after having located the suitable parameters in core locations immediately after the RTJ GETLJD instruction. If the status is N GETLJD is called with Q = 0 (for enter new job). Else GETLJD is called twice: once with Q = 1 (for locate job), after which a check is made if the maximum record length specified in the command is at least as large as the record length in the current LJD entry. If this condition is met a second call is made, with Q = 4 (for modify existing entry).

INTERFACE

1. INPUT PARAMETERS OR CONDITIONS

The parameters for EJB are given by the user when entering an EJB command.

<u>Syntax</u>

EJB,a(1),aaaaaa(2),nnn(3),aaa(4),nnn(5),nnnn(6)   CR

(1) Job Status - <u>N</u> (new) or <u>R</u> (revised).

(2) Job Name.

(3) Magnetic tape format (1 through 255).

(4) Verification type - <u>F</u> (format), <u>E</u> (error field), and/or
    <u>U</u> (unbalanced field), <u>N</u> (no verification type).

(5) End of batch own code number (0 through 127).

(6) Maximum record length for job (1 through 1000).
    In case of status <u>R</u>, the max. record length may be increased but
    may not be reduced.

I = current ATT address.

2. EXIT METHODS

   If no ERROR is detected RETURN to SV.

   If an ERROR is detected exit is made to REJSUP.

3. OUTPUT PARAMETERS OR CONDITIONS

   I = current ATT base address

   The error message which might be printed are:

   a. INVALID JOB NAME

       The job name started with a blank or job name was more than 6 characters.

   b. INVALID OWN CODE NO.

       End of batch own code greater than allowed (255).

   c. UNDEFINED JOB NAME

       Name of job to be modified not in LJD.

   d. INCOMPLETE COMMAND

       Command does not include all required parameters.

   e. INVALID COMMAND FORMAT

       Missing comma before parameter list.

   f. INVALID RECORD LENGTH

       The maximum record length exceeded the maximum allowed or an
       attempt was made to modify the maximum length in an existing job
       to less than before.

   g. INVALID MAG. TAP FMT NO.

       Magnetic tape number was less than 0, or greater than 255.

   h. INVALID VERIFICATION OPTION

       The characters given for the verification option were not N, F, U, or E,
       or the same letter appeared more than once, or no specification given.

   i. MAX. NO. OF JOBS EXCEEDED

       Job directory already full.

   j. DUPLICATE JOB NAME

       The new job to be entered had a name existing already in the LJD.

   k. INVALID STATUS OPTIONS

       Status was not N or R.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK ($2)
        NZERO ($12)
        ONEBIT ($23)
        ZEROBIT ($33)
        SST ($47)
        EXTSST ($47)
        GETBIN ($99)
        GETALP ($9B)
        GETCOM ($9C)
        SUPPW ($97)
        SV ($B1)
        REJSUP ($B2)

    b. ATT

        SUPTMP (36+10)
        SAVJOB (SUPTMP+1)
        MRL (SUPTMP+4)
        EOBOC (SUPTMP+5)
        JIDX (3)

    c. SST

        STEMP (20)

    d. EST

        MAXOCD (4)
        MAXMTF (4)
        MAXRCL (38)

    e. CONSTANTS (ASCII)

        Y ($59)
        N ($4E)
        E ($45)
        U ($55)
        R ($52)
        V ($56)
        F ($46)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

    GETLJD   -   to enter or modify a job entry
    SETMOV   -   entry in ENEXSR to enable the program to move
    GETALP   -   entry in SCANV to pick up nonnumeric parameters
    GETBIN   -   entry in SCANV to retrieve numeric characters
    GETCOM   -   entry in SCANV to check for comma
    REJSUP   -   entry in SUPER for error exits
    SV   -   entry in SUPER for normal exit

4. DATA AREAS USED

    ATT
    SST
    EST
    LJD

PHYSICAL REQUIREMENTS

1. MEMORY SPACE REQUIREMENTS

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME____EMM - Display and Store Character in Entry Mode_____LEVEL 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    EMM has 14 entry points:

    a. EMM moves the current character to the Terminal Record Buffer, updates character counter checks for end of field, and continues processing at EMMS1.

    b. EMMS checks for signed numeric, and makes sure a sign exists where necessary.

    c. EMMSKP sets up the requirement for the SKIP key to be next, and enters IDK mode.

    d. EMM1 performs validation checks on the current field and continues processing at EMM2.

    e. EMM2 exits to VMM if the current field is verifiable and part of an inserted record, exits to CLRSR if in Pseudo Entry mode and not an inserted record, and continues processing at TSTREC otherwise.

    f. TSTREC continues processing at TSTODD if the current field is the last field in the format. If not, TSTREC advances to the next field in the format and continues processing at EMM5.

    g. EMM5 performs automatic SKIP/DUP functions as required for the current field.

    h. TSTODD pads the current record to an even number of characters, if necessary, by adding a zero byte to the record, exits to CLRSR if the current record is an inserted record, and exits to EORENT if not.

    i. VALDTE performs end of field validation: Nonzero Test, Limits Test, Special Test, Own Code, and Counter updates.

    j. COUNT adds/subtracts the value of the current field to/from the designated counters, if any.

    k. BOFFE advances pointers to the next field in the Format Table and stores data type of next field in Active Terminal Table (ATT).

    l. SETBOF inserts data type of new field in ATT and displays field number.

    m. STTYPE moves the Data Type code from the Current field description to the ATT.

    n. AUTO performs automatic functions on a field, as required.

2. DETAILED DESCRIPTION

   a. EMM

      1. (EMM). Call SBYTE to store current character in Terminal Record Buffer (TRB).

- Increment ATT/FCNT by 1 (no. of characters in field) and ATT/RCNT by 1 (no. of characters in record).

- Call DISPLY to display current character and current column number.

- Check for end of field. If not, exit to CLRSR.

   b. EMMS1

      1. (EMMS1). If end of field, check data type. If not signed numeric field, go to step 2.

- If last character of signed numeric field is plus or minus sign, go to step 2.

- Back up ATT/FCNT and ATT/RCNT by 1, and call DISPLY to blank out last character in display.

- Set ATT/SW 2/4 (Dup must recall data from disk).

- Exit to ERROR with (Q) = A2 (SIGN MISSING).

      2. (EMMSK1). If field boundary checking is selected for this field, go to step c.1. If not, go to step d.1.

   c. EMMSKP

      1. (EMMSKP). Set ATT/SW1/9 = 1 (SKIP key must be next).

- Set ATT/MODE/14 = 1 (IDK mode), and exit to CLRSR.

   d. EMM1

      1. (EMM1). Call VALDTE to perform validation checks on the field.

   e. EMM2

      1. (EMM2). If ATT/SW1/12 = 0 (not insert), go to step 2.

- If in Entry mode, go to step f.1.

- If positioned beyond last verified record, go to step f.1.

- Call CHKVRY to check whether current field is to be verified. If not, go to step f.1.

- Set ATT/MODE/12 = 1 (Pseudo Verify mode), and exit to RVFVER (in VMM) to verify the field.

AA5641

PRINTED IN THE USA

2. (EMM3). If not Pseudo Entry mode, go to step f.1.

3. (PSEUDO). Set ATT/MODE/13 = 0 (not Pseudo Entry mode) and
    ATT/MODE/14 = 1 (IDK mode).

    ● Set ATT/SW1/11 = 1 (COR key required), and exit to CLRSR.

f. TSTREC

1. (TSTREC). If current field is last field in format, go to step h.

    ● If not, call DISPLY to set display position to next field.

    ● Call BOFFE to select next field description in format and put
      data type of next field in ATT/DT/13-15.

g. EMM5

1. (EMM5). Call AUTO to perform automatic SKIP/DUP functions as
    required, and exit to CLRSR.

h. TSTODD

1. (TSTODD). If the current record contained an odd number of
    characters, call SBYTE to add a binary zero character to the record.

    ● Clear ATT/SW2/4 (no need to read from disk for duplication).

    ● If ATT/SW1/12 = 1 (Record insertion), go to step e.3.
      If not, exit to EORENT.

i. VALDTE

1. (VALDTE). Call ENTSR to save return address.

    ● Set ATT/SW2/13 = 0 (No revalidation required).

2. (TSTNZ). If Nonzero Test is specified, call NZT to test the field
    for all spaces or all zeros.

3. (TSTLL). Call FMTPAR to locate Limits parameters. If Limits Test
    is specified; call LUL to perform the test.

4. (TSTST). Call FMTPAR to locate Special Test parameter. If Special
    Test is specified, fetch address of Special Test routine from
    STJLNG table. If the address indicates an unpatched External
    Reference, exit To ERROR with (Q) = $9 (Special Test Error). If not,
    call the designated Special Test routine to perform the test.

5. (TSTOC). Call FMTPAR to locate Own Code parameter. If Own Code
    routine is specified, call ENTPRG to load and execute Own Code,
    with returns to step 6.

6. (TSTCNT). Call FMTPAR to locate Batch Counter parameters. If counters are specified, call COUNT to add or subtract the contents of the current field to the designated counters.

- If field was marked as erroneous but now passes validata, go to FLAGS (3rd entry point of REPKEY) to clear the indicators.

- Exit to EXTSR to return to caller.

j. COUNT

1. (COUNT). Save counter number in ATT/ATEMP + 1/8-15. Call PACK to pack current field for arithmetic purposes.

2. (COUNT2). Shift ATT/ATEMP +1/ left 4 store pack and check 4 L.S.B. If counter number = 0 (counter not defined), go to step 3.

- Calculate address of counter in ATT. If addition is specified, call ADD to add packed value of current field to designated counter, and continue at step 3.

- If subtraction is specified, call SUB to subtract the packed value of the current field from the designated counter.

3. (CHECK). If counter number in ATT/ATEMP + 1/12-15 ≠ 0 (second counter number 0) repeat step 2 for the second counter. Otherwise exit to the caller.

k. (BOFFE)

1. (BOFFE). Update SST/FFA to point to next field in Format Table.

- Update ATT/RFA/0-7 to relative address of next field in Format Table, and increment ATT/RFA/8-15 by 1.

- Call STTYPE to move data type of next field to ATT/DT/13-15.

- Exit to caller.

l. (SETBOF)

1. (SETBOF). Call ENTSR to save return address.

- Call STTYPE to move data type of current field to ATT.

- Call DISPLY to display new field number.

- Exit to EXTSR, which returns to caller.

m. (STTYPE)

1. (STTYPE). Move data type from current field description in Format Table to ATT/DT/13-15, and exit to caller.

n. (AUTO)

1. (AUTO). Call ENTSR to save return address.

- Set ATT/ECODE = $FFFF, to indicate whether column display is needed.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____EMM - Display and Store Character in Entry Mode_____LEVEL 6

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___EMM - Display and Store Character in Entry Mode___   LEVEL 6

INTERFACE

1. ENTRY METHOD

    EMM, EMMSKP, EMM1, EMM2, TSTREC, EMM5, EMM1S and TSTODD are all entered
    using the JMP instruction, with no special parameter requirements.

    VALDTE, BOFFE, SETBOF, STTYPE, and AUTO are closed subroutines, that is,
    they are entered using the RTJ instruction. No parameters are required.

    COUNT is a closed subroutine with a parameter requirement.

        (A) = abcde

        Where:

                a (bit 15) = 0 means add the field to first counter.

                            = 1 means subtract the field from first counter.

                b. (bit 12-14) = first counter number.

                c. (bit 11) = 0 means add the field to second counter.

                            = 1 means subtract the fields from second counter.

                d (bits 8-10) = second counter number.

                e (bits 0-7) = 0

        In each case, (1) = Base address of Active Terminal Table (ATT).

2. EXIT METHOD

    EMM, EMMSKP, EMM1, EMM2, TSTREC, EMM5, and TSTODD exit to CLRSR, EORENT,
    or ERROR.

    VALDTE, BOFFE, SETBOF, STTYPE, AUTO, and COUNT all return control to the
    Caller under normal conditions. VALDTE exits to ERROR if the SPECIAL
    TEST routine is undefined. No special parameters are returned.

    (1) = Base address of ATT.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___EMM - Display and Store Character in Entry Mode___   LEVEL 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        ENTSR ($91)
        EXTSR ($92)
        CLRSR ($93)
        GBYTE ($94)
        SBYTE ($95)
        ERROR ($96)
        ENTPRG ($9E)
        DISPLY ($A0)

    b. SST

        FFA (12)
        STEMP (20)

    c. ATT

        CURCH (2)
        MODE (2)
        FCNT (3)
        RCNT (4)
        ATRB (6)
        DT (13)
        RFA (14)
        BMOD (15)
        SW1 (19)
        SW2 (20)
        RCORD (31)
        RCD (32)
        ECODE (34)
        ATEMP (73)

    d. ERROR CODE

        ERRA2 (2)

AA5641   PRINTED IN THE USA

2. PROGRAM DEPENDENCIES

ENEXSR	{ ENTSR	Save return address in Exit stack
	EXTSR	Exit to last return address in Exit stack
	CLRSR	Clear Exit Stack and function lock
GSE	{ GBYTE	Get current character from Terminal Record Buffer
	SBYTE	Store character in TRB
	ERROR	Display error code
	ENTPRG	Enter a disk resident program
	DISPLY	Display routine
VMM	{ CHKVRY	Check whether current field should be verified
	RVFVER	Verify current field
EMM	BOFFE	Advance to next field
	EORENT	Write current record to disk
	FMTPAR	Locate field parameter in Format Table
ARITH	{ NZT	Nonzero test
	LUL	Limits test
	PACK	Convert current field to packed format
	ADD	Addition routine
	SUB	Subtraction routine
STJ	(STJLNG)	Special test routine
SKPDUP	{ DUPIT	Advance counters to end of current field
	SKIPIT	Perform field-skipping operations

3. DATA AREAS USED

SST	System Status Table
ATT	Active Terminal Table
TRB	Terminal Record Buffer
FMT	Format Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$154_{16} = 340_{10}$ words.

2. RELOCATION TYPE

Core resident, loader relocatable.

3. ENTRANCY CLASSIFICATION

Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____EMT - Entry Mag-Tape Format_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    EMT sets up the MFT table parameters according to the ones entered by
    the supervisor. Every data written on tape has to be handled by a
    corresponding format.

2. DETAILED DESCRIPTION

    EMT, like all other supervisor functions, starts by allowing itself to
    be moved (SETMOV). Then it looks for the comma following the EMT
    command. If not found, the command is rejected and a corresponding
    error message is sent. If found the program proceeds by checking the
    input parameters against a table containing every possibility. ASCSUB
    checks the first parameter (status). If legal and followed by comma,
    the format number is read. Illegal number results in an error message
    and rejection of the format. If legal number, it is compared with the
    limits (1 to 255). When it doesn't lie between them it is considered
    illegal, and the command is rejected.

    Otherwise, the current MTF (5 words) is read from the disk (by subroutine
    SUBR with $Q = 0$). The A/I bit is compared to bit 15 of MTFBUF + 5 (set by
    ASCSUB if new format). Both 'active' and 'new' or 'inactive' and 'revise'
    attributes result in an error.

    If no such error is detected, the A/I bit is stored in word 3 and the
    other words of MTFBUF are cleared.

    Then, the recording mode, recording code, and overpunching parameters are
    read. The ASCSUB subroutine sets the corresponding bits in MTFBUF. Any
    errors (illegal parameter, missing comma) cause an error to be signalled.

    The blocking mode is checked for correctness, in relation to other
    parameters. Variable field recording is illegal if blocking mode is
    fixed or unblocked. If blocking is variable, BCD code is illegal. And
    only if blocking mode is 'U', may the end-of-record code follow it. Every
    error here causes the command to be rejected. The 'U' attribute causes
    the built format to be written on disk and control returns to supervisor.
    The block length is retrieved and checked. If it is 0, and blocking mode
    is 'U', or the length lies between the limits (18 to 4000) and is even,
    then it is stored in MTF-W1.

The record length must follow if fixed recording, otherwise an error is raised. If zero record length is specified, the recording mode should not be fixed. If not zero length, it is written in MTFBUF-W2 and the recording mode is checked to be fixed. If not fixed, or the block length is not a multiple of the record length, an error message is sent.

If no more parameters are passed (no padding) the record length is checked to be greater than 18 and the command is either accepted or rejected, accordingly.

PAD gets the padding character and sets MTFBUF bit P if binary padding is needed. If no padding, the blocking mode should be either not fixed or greater than 18; other configurations are illegal. If ASCII padding is needed, one character is read and stored in CHAR. If binary padding the recording mode should not be BCD, otherwise an error is sent.

Two characters are read, checked to be 0 - 9 or A - F, and their hexadecimal equivalent is stored in CHAR, then in MTFBUF-W0.

If MTF own code follows it is read and checked to be between 0 and 31, otherwise it is illegal. If correct it is stored in MTFBUF-W0.

If a file label number follows it is read and checked to be in the correct range (1 to 31). If it is, that number is inserted in MTFBUF-W1. Next, the label parameters option bit is set (if requested) by ASCSUB, in MTFBUF-W3. And lastly, the batch grouping factor is read (if specified), checked to be in the range of 1 to 999 and stored in MTFBUF-W3. If all the parameters were correctly specified, the resulting MTFBUF is written on disk by SUBR with Q = 1 as parameter. SUBR gets the right disk address for that MTFBUF and the writing is done through the WRTDWA subroutine.

ASCSUB sets bits in MTFBUF according to the input parameter (obtained through GETALP). A printer to the set up-table is increased after each check and a new control word is obtained from the table accordingly. Each control word has the form:

N1/a, N3/b, N4/c, A8/d and its interpretation by ASCSUB is:
set bit c in word b of MTFBUF if the input character is d; a is 0 except for the last possibility in that field, there is 1.

CØMEØR is a subroutine to check the end-of-record condition. If the next character is a comma, CØMEØR returns with Q = 0. If any ASCII character follows, returns with Q = 1. If end-of-record is detected (possibly preceded by blanks), the recording mode has to be fixed (otherwise blocking mode should follow) and then the resulting MTF is written on disk and control passed to SV. In any other case, the instruction is rejected, after sending an error message.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___EMT - Entry Mag-Tape Format_____Level 6

INTERFACE

1. ENTRY METHOD

    Called by DSKSUP through ENTPRG.

2. INPUT PARAMETERS

    I = base address of ATT.

3. EXIT METHOD

    If any errors detected exit to REJSUP with a corresponding error message.
    If no errors, exit to SV.

4. OUTPUT PARAMETERS

    (I) is unchanged. New (or revised) MTF written on disk.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LØCØRE

        LPMASK (2)
        NZERO ($12)
        ZROBIT ($33)
        ONEBIT ($23)
        ZERO ($22)

    b. SST

        SST ($47)
        STEMP (20)
        STEMP1 (21)
        STEMP2 (22)
        STEMP3 (23)
        EXTSST (0)
        MTFADD (42)

    c. TRANSFER VECTOR

        CLRSR ($93)
        GETBIN ($99)
        GETALP ($9B)
        GETCOM ($9C)
        REJSUP ($B2)
        SV ($B1)
        SUPRW ($97)
        CLCDWA ($90)
        REDDWA ($A9)
        WRTDWA ($AA)
        EXTSR ($92)
        ENTSR ($91)

d. SUPERVISOR ATT

SUPCOM (36)
SC (46)
MTFBUF (36)
TEMP1 (42)
TEMP2 (43)
PADCH (44)
COUNT (45)
MTFN (46)
WORD (47)
FLAG (48)
CHAR (49)
SAVPNT (50)
BLKLNG (51)

e. DATA AREAS USED

ATT   EST   Interrogate Buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$19B_{16} = 411_{10}$ words.

2. RELOCATION TYPE

Loader relocatable, run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____ENEXSR - Common Entry/Exit Routine_____Level 6_

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    ENEXSR has the following seven entry points to handle various entry and
    exit conditions.

    a. ENTSR saves in the ATT exits stack the return addresses of subroutines
        with I/O or retry.

    b. RTNSR is entered by return jump and exits to the caller with Q=last
        address in the ATT exits stack.

    c. EXTSR is entered by direct jump and exits to the last address in the
        exits stack.

    d. CLRSR clears the ATT exits stack, does all necessary housekeeping to
        release core and reinitialize the ATT for the next function, and exits
        to the dispatcher.

    e. SETMOV clears the I/O bit in the header of a disk resident program to
        enable moving the program in core.

    f. STRTRY is used to delay a function for one terminal processing cycle
        after which control is returned to the address in A upon entry.

    g. WAIT is entered by return jump and does the same thing as STRTRY except
        control is returned to the caller.

2. DETAILED DESCRIPTION

    a. ENTSR saves A and Q and looks for an entry in the ATT exits stack.
        If no entry is found, a MONITOR error 2 is generated and the ATT
        is function locked. If an entry is found, it converts the return
        address to relative if ATT/PI/15 is set clears ATT/PI/15, and sets
        the relative entry bit.

        If ATT/PI/15 is not set, the return address saved is absolute. A and Q
        are restored and control is returned to the caller.

    b. RNTSR saves A and temporarily clears ATT/PI/15 (relative addressing).
        It finds the last nonzero entry in the exits stack, temporarily saves
        that exit address and clears the stack of that entry. It checks the
        relative bits for relative addressing and adjusts according to where
        that program is loaded if the address was relative. It resets ATT/PI/15,
        sets Q equal to the last exit address, restores A and returns control to
        the caller.

    c. EXTSR detailed processing is the same as RTNSR except that it is entered
        by direct jump and exits to the last address in the exits stack.

d. CLRSR clears each entry in the exits stack, ATT/LOCK/14 (function lock), and ATT/TRMNT/15, 14 (terminate function bits). If ATT/PI/15 (relative address) is set, it releases the disk function via EXDFUN. If an interrogate buffer is in use, it clears the interrogate buffer user count and ATT/IRB (interrogate buffer address). It sets SST/SLOCK/SM (storage move required bit) and exits to the dispatcher.

e. SETMOV gets the program directory index (PRD) to find the address where the program is loaded in order to clear the I/O bit in word 1 of the program header. (See analytical supplement.)

f. STRTRY return jumps to ENTSR to store the return address in the exits stack. It sets ATT/LOCK/13 (pseudo function lock) and exits to the dispatcher.

g. WAIT is the same as STRTRY except that it is entered by return jump and returns control to its caller.

INTERFACE

1. ENTRY METHOD AND INPUT PARAMETERS OR CONDITIONS

    In all cases, I = ATT base address.

    a. ENTSR is entered by a return jump which must be the first instruction after the entry cell of the program.

    b. RTNSR is entered by return jump.

    c. EXTSR is entered by direct jump.

    d. CLRSR is entered by direct jump.

    e. SETMOV is entered by return jump.

    f. STRTRY is entered by direct jump with A = return address to be stored in ATT exits stack.

    g. WAIT is entered by return jump so that the caller's return address is stored in the ATT exits stack.

2. EXIT METHOD AND OUTPUT PARAMETERS OR CONDITIONS

    a. ENTSR returns to the caller with A, Q and I saved.

    b. RTNSR returns to the caller with Q = last address of exits stack and A, I saved.

    c. EXTSR exits to last address in exits stack with A, Q, I saved.

    d. CLRSR exits to the dispatcher.

    e. SETMOV returns to the caller with A, Q destroyed and I saved.

    f. STRTRY exits to the dispatcher.

    g. WAIT exits to the dispatcher.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __ENEXSR - Common Entry/Exit Routine__   Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. SST

        SLOCK (1)
        PRDIR (3)
        SST ($47)
        MONITR ($8C)
        EXDFUN ($9F)

    b. ATT

        PI (21)
        TRMNT (23)
        IRB (27)
        ENTRY (79)
        RELENT (78)

    c. LOCORE

        LPMASK (2)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)

    d. CONSTANTS

        STACKL (8)   Length of exits stack.

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

    NDISP (Dispatcher)
    ENTPRG (EXDFUN entry point to exit from a disk function.)
    MONITR (Debug routine)
    ENTRSR ($91)

4. DATA AREAS USED

    SST   System status table
    ATT   Active terminal table
    PRDIR   Program directory
    IRB   Interrogate buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $7E_{16} = 126_{10}$ words.

DOCUMENT CLASS   CYBERDATA IMS

PROGRAM NAME   ENEXSR - Common Entry/Exit Routine   Level 6

2. RELOCATION TYPE

Loader relocatable.

3. ENTRANCY CLASSIFICATION

Reentrant.

ATTACH ANALYTICAL SUPPLEMENTS

Disk Resident Program Header

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Backwards Pointer																	0
I/O	Code				Length										1
Interval							User Count							2
Disk Address																	3
Time																	4

AA5641   PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME____ENTPRG - Control for Disk Resident Programs_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   The program ENTPRG has two entry points: (1) ENTPRG is called to access a disk resident program, and (2) EXDFUN is called to release a disk resident program. All disk resident programs in Cyberdata are accessed and released via ENTPRG.

2. DETAILED DESCRIPTION

   ENTPRG calls the common entry subroutine to save the return address in the ATT exits stack. The entry parameters in A and Q are saved temporarily in ATT/ATTMP3 and ATT/ATTMP2. If the request is for a key function routine, the key function code is adjusted to be a proper index into the program directory. Next the displacement from the beginning of the PRD to the entry for the requested program is calculated and saved. The subroutine GETBIT calculates the word, bit and address of the PRD control block where the status of the requested program is saved. If the program is already in core, the I/O bit is set for the block of core the program occupies and the user count is incremented by one.

   (COMMON). The caller's request parameters are checked for load only or load and go. If the request was for load only, Q is set to the program address (backwards pointer) and control is returned to the caller via the common exit subroutine. If the request was for load and go (LOADGO), a check is made to see if the entered routine is a subroutine. If it is not, the last address is removed from ATT exits stack. The function index for the requested program is stored in ATT/FUNIDX. The address within the program of the entry point requested by the caller is calculated and stored in SST/TEMP, then that address is jumped to.

   If the requested program is not already in core, a check is made to see if it is currently being loaded into core (in which case it is handled as an in core case after a delay while it's load is completed). If it is not in core and not being loaded into core, a check is made to see if the program exists on the disk. If it does not exist on the disk, a request to load a dummy function is made. The calling parameters are set up for a read of one word of the program to be read into ATT/ATTMP5. This first word contains the program length which is increased by one for the back- wards pointer and used as input to CAM for a core request. If CAM does not have the required amount of core, control is returned to the caller via the common exit subroutine or to REJSUP if the request was a supervisor function or retry is executed according to the caller's parameters. When core is obtained, the I/O bit in the program buffer is set, the PRD address is cleared to indicate that the program is currently being loaded, and the parameters are set up for the disk read by PIO (REDSEC). If the caller's terminal is not in supervisory mode, the request priority for the read is incremented by one. After the program is loaded, the backwards pointers are set in the PRD and program buffer, the PRD control word bit for the requested program is set, and the logic at (COMMON) is followed from here.

EXDFUN decrements the user count of the program buffer. If the user count is not then zero, the time factor in the program buffer is updated, ATT/FUNIDX is cleared, memory move bit (SST/SLOCK/14) is set and control is returned to the caller. If the user count is zero and timing was not specified, ATT/FUNIDX is cleared, the I/O bit in the program buffer is cleared, the program buffer is released by CAM (RPRD), SST/SLOCK/14 is set and control is returned to the caller.

INTERFACE

1. ENTRY METHOD

    Return jump

2. INPUT PARAMETERS OR CONDITIONS

    ENTPRG:

    1 = current ATT base address

    Q = bit 15 = 1, load only

    = 0, load and go

    bits 14-12 = 0, program is miscellaneous

    = 1, program is a key function

    = 2, program is an interrogate function

    = 3, program is a supervisor function

    = 4, program is a mag tape function

    = 5-6, undefined

    = 7, program is an own-code

    bit 11   = 1, wait until core available
                = 0, if core not available return to caller with
                     Q = 0 or to REJSUP if supervisor function

    bit 10   = 1, entered routine is a subroutine

                = 0, entered routine will not return

    bits 6-0   = program number of current group

    A = bits 7-0   = the entry point number in the requested program

    NOTE: All ATT temporaries except for the first one are used by
           the program

    1 = current ATT base address

    ATT/FUNIDX = PRD index of program to be released

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____ENTPRG - Control for Disk Resident Programs_____Level 6

3. EXIT METHOD

   ENTPRG: Discussed in detailed description and input parameters section.
   EXDFUN: Return to caller.

4. OUTPUT PARAMETERS OR CONDITIONS

   ENTPRG: Discussed in detailed description and input parameters sections.
   EXDFUN: User count is decremented and the program area is released if
           necessary. FUNIDX is cleared and Control is returned to caller.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)
      SST ($47)
      TRACE ($8B)
      MONITR ($8C)
      ENTST ($91)
      RTNSR ($98)
      REDSEC ($A7)
      CAM ($AE)
      WAIT ($B0)
      REJSUP ($B2)
      DISPCH ($EA)

   b. SST

      SLOCK (1)
      TMFCT (2)
      PRDA (3)
      PRINCR (13)
      CLOCKS (17)
      TEMP (20)
      TEMP1 (21)

   c. ATT

      FUNIDX (21)
      ATTMP2 (74)
      ATTMP3 (75)
      ATTMP4 (76)
      ATTMP5 (77)

   d. EXTERNALS

      ABSADD entry point in PIO indicates that the next entry to PIO is of
      absolute type address.

2. MACROS USED

   None

3. PROGRAM DEPENDENCIES

PIO	REDSEC	Read section or part of it
	ABSADD	
CAM		Core allocator module
ENEXSR	ENTSR	Enter subroutine
	EXTSR	Exit subroutine
	RTNSR	Return address from stack
	WAIT	Wait one cycle
NDISP	DISPCH	Dispatcher
MONITR	MONITR	Debugging module
	TRACE	Debugging module
SUPER	REJSUP	Reject supervisor function

4. DATA AREAS USED

   ATT
   SST
   PRD

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $C8_{16} = 200_{10}$ words.

2. RELOCATION TYPE

   Loader relocatable.

3. ENTRANCY CLASSIFICATION

   Reentrant code.

AA5641

PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___ENTRY - Entry Mode Processor_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The ENTRY program processes the interrogate function

                    ENT,JJJJJJ,BBB,000,AAAA

    Where: JJJJ   is the Job name (maximum 6 char.).
            BBB   is the batch number (1-999).
            000   is the operator number (000-999).
            AAAA   is the autosequencing count (0000-9999, assumed to be 0000,
                    if missing).

2. DETAILED DESCRIPTION

    ENTRY first calls SETMOV to clear the I/O bit in the program header so
    the program can be moved in core. If the terminal is not in Idle mode,
    an E1 (invalid request) error is generated via ERROR. The ATT is reinitialized
    for the start of a new batch. The job name is checked first for no more
    than six alphanumeric characters. If not, a D4 (illegal job name) error
    is generated via ERROR. A comma must follow the job name. The batch number
    is converted from ASCII to binary via SCANV (GETBIN). If the batch number
    is not a numeric value in the range of 1-999 a D5 (illegal batch number)
    error is generated. The operator number is then converted to binary and
    checked for a numeric value less than 1000. If not, a D1 (invalid operator
    number) error is generated via ERROR. The autosequencing number, if intended,
    is also converted to binary and checked for a numeric value less than 1000.
    If it failed the test, a D6 (invalid autosequence count) is generated.

    The batch number and operator number are converted to BCD and stored in
    the ATT. In case the display type of the station is a CRT the job name is
    saved in the output buffer for further display requirements. GETLJD is
    called to ensure that the job name entered is a legal one. The amount
    of core needed for the maximum record length plus header is allocated via
    CAM. If no core is available, an F4 (Memory full-try again) error is
    generated. The allocated core is used for a TRB whose pointers are initialized.

    Next the batch is assigned to a disk. In a multiple disk system, the disk
    assignments are made once to each disk each cycle, except that the system
    disk is passed over every other cycle. DAM (GITRT) is called to assign
    one logical track to this batch. If there is no disk space available,
    the TRB area is released and an F3 (disk full) error is generated.

The first track of the batch (assigned by GITRT) is stored in the ATT. SPILL (Active Batch Table handler) is called in to insert this batch in the Active Batch Table. At this point à check is made for a duplicate batch number. If one is found, the allocated disk track is released via DAM (RELTRT), the TRB area is released and a D5 (invalid batch number) error is generated. BORENT (LINKTR) is called to set up the track table and write it to the first sector of the batch. The start time of current run, accumulated time of run, start time of batch and operator number are saved in the first track of the batch. The available number of words in the track, the word address of the first record to be written on the disk, and the ATT are recorded on the disk. Before the ATT is recorded on disk, it is set so that document or format select must be keyed in and the mode of the ATT is put to be in Entry. DISPLY is called to set display definition bits and regenerate the display. The program releases its core via EXDFUN and exits to the dispatcher via CLRSR.

INTERFACE

1. ENTRY METHOD

    ENTRY is a disk-resident program loaded via ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.
    AINT,I = pointer to the parameters entered (user count of the buffer = 0).

3. EXIT METHOD

    In normal conditions: Release core via EXDFUN.
                            Exit to dispatcher after clearing exits stack via CLRSR.
    In error conditions: Exit to ERROR.

4. OUTPUT PARAMETERS OR CONDITIONS

    New batch is initialized.
    I = current ATT base address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        CLCDWA ($90)
        CLRSR ($93)
        ERROR ($96)
        GETBIN ($99)
        GETALP ($9B)
        GETCOM ($9C)
        EXDFUN ($9F)
        DISP ($A0)
        SPILL ($A1)
        WRTDWA ($AA)
        WRTATT ($AC)
        CAM ($AE)

    b. SST

        EXTSST (0)
        CLOCLM (16)
        CLOCK (17)
        STEMP (20)

    c. ATT

        FTOB (1)
        CMODE (2)
        JIDX (3)
        ASCNT (5)
        TRBA (6)
        AVLTK (8)
        TOPR (9)   Operator number (saved temporarily).
        TJOB (10)   Job name   (saved temporarily).
        TSBCH (11)   Batch number   (saved temporarily).
        TCLK (11)   Clock value   (saved temporarily).
        NDA (11)
        TACTME (12)   Accumulated time (saved temporarily).

RFA (14)
BMOD (15)
SW1 (19)
FUNIDX (21)
ABTIDX (22)
AINT (27)
ATOPT (33)
BATCH (35)
ATEMP (73)
ATEMP1 (74)
ATEMP2 (75)

d. EST

DSKST (29)

e. ERROR CODES

CD1 ($D)	Invalid operator number
CD4 ($10)	Invalid job name
CD5 ($11)	Invalid batch number
CD6 ($1D)	Invalid autosequence count
CE1 ($12)	Invalid request
CF3 ($16)	Disk full
CF4 ($17)	Memory full-try again

f. EXTERNALS

SETDIS	Parameter for DISPLY to set screen location according to A register.
REGEN	Parameter for DISPLY to regenerate the display.
SETMOV	
DB	
GETLJD	
GITRT	
RELTRT	
LINKTR	

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

SCANV	(GETBIN)	Get the number from input string in binary form.
	(GETALP)	Get an alpha character from input string.
GSE	(ERROR)	Display error at terminal and terminate the function.
CAM		Allocate core for the TRB.
DAM	(GITRT)	Allocate one logical disk track for data storage.
	(RELTRT)	Release the logical track assigned.
SPILL		Insert batch in active batch table.
PIO	(CLCDWA)	Calculate disk word address.
	(WRTDWA)	Write in disk-word addressing form.
	(WRTATT)	Write ATT to disk.
ENEXSR	(CLRSR)	Clear exits stack and go to dispatcher.
	(SETMOV)	Enable the disk resident program to be moved.
DISPLY-		Set data definition bits and regenerate display.
ARITH	(BD)	Binary to decimal conversion.
GETLJD		Validate job name and get LJD index.
BORENT	(LINKTR)	Set up track table and write to first sector on disk.
ENTPRG	(EXDFUN)	Release the disk resident program.

4. DATA AREAS USED

    ATT
    SST
    TRB
    DTL
    TOPT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $118_{16} = 280_{10}$ words.

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant code.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___EOB - End of Batch Processor___ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The End of Batch Processor checks batch counters for zero values, calls
    the EOB Own Code routine, if there is one, sets the last record flag in
    the batch, and builds and writes a statistics record.

    There are two entry points in EOB: EOB is the beginning of the EOB procedure,
    ENT2 is a continuation entry point for Own Code return and erroneous batch
    counter return. If a batch counter is not zero, the program TOTAL is
    called to display the message COUNTER xx = nnnn... where xx=counter
    number and nnnn = value of counter.

2. DETAILED DESCRIPTION

    a. EOB

        1) (EOB). The 4th character in the input string is not EOR, or if
           number of records in batch = 0, or if in read mode, or if not at
           end of data (ATT/SW2/11=0) in verify mode, or if ATT/RCNT ≠ 0
           (character count in current record), then set ATT/SW1/8 = 1
           (reset must regenerate display), set (Q) = $12 (Illegal Request)
           and exit to ERROR.

        2) (IFBOR). Set ATT/BMD1 = 0 (internal flag to indicate return) and
           ATT/25 = 0 (internal flag to indicate balancing).

           (NXTCTR). Check the batch counters in ATT for zero values. If
           all values are zero, go to step b.3.

        3) (BALERR). When a batch counter is nonzero, save the counter number
           in ATT/BMD1 and ATT/25; and set ATT/SW1/0 = 0 (EOB Validation Error).
           Call EXDFUN to release EOB program. Call ENTPRG to load and execute
           TOTAL.

    b. ENT2

        1) (ENT2). If ATT/BMD1 0 (Own Code return), set ATT/SW1/0 = 0
           (EOB Validation Error) and go to step b.4.

        2) (IFLC). Convert next batch counter number to an index to batch
           counters and go to step a.3.

        3) (BAL). Call GETLJD to locate job in Legal Job Directory (LJD).
           If EOB own code specified, set ATT/BMD1 = $FFFF, set ATT/SW1/0 = 1,
           call EXDFUN to release EOB program, and call EOBIT to load and
           execute own code program. EOBIT will load and execute EOB when
           own code program is finished.

AA5641

4) (NOOC). Call READN to read last record of batch from disk, set TRB/5/15 (Last Record Flag) and call REWRIT to rewrite last record of batch.

5) (NOENT). Call FRZATT to write updated ATT record to disk.

6) (GENJN). Call GETLJD to locate job in LJD.

7) Set up statistics record in ATT/34 through ATT/49 as follows:

ATT/34 = SRL/0		Mode/Verif. Type + Status/Job Index*	
35	1	Batch number	already in ATT/35
36	2	Operator Number	
37	3	Start Time	from Data Track header
38	4	Date	from ADAYTO and ANONTO in SYSDAT
39	5	Accumulated Time	calculated**
40	6	Number of Key Strokes	from ATT/KEYDPR
41	7	Number of Verify Corrections/Verifiable Records	from ATT/VR or ATT/VRCC
42	8	Number of Erroneous Records	from ATT/RCDER
43	9	Number of Records from last SRL written	from ATT/RCD
44	10	Year	from AYERTO in SYSDAT
45	11	Unassigned	
46	12	Unassigned	
47	13	Job	
48	14	Name	from LJD
49	15	(6 characters)	

* Mode (bits 13-15) = 0 if Entry Mode
Verif. Type + Status (bits 8-11)

		bit 11:	Verify according to format.
Verification		bit 10:	Verify erroneous fields.
Type		bit 9:	Verify fields with unbalanced registers.
Status		bit 8:	Batch unbalanced.

** Accumulated time = SST/CLOCKS - ST + AT
ST = Start time for current run
AT = Accumulated time for batch   from Data Track header

8) Call SPILL to update the mode of the batch in the Active Table.

9) Call WSS to write record in statistics file.

10) Update record count and accumulated time in the DTL.

11) Call DISPLY to clear display and set display to Idle mode.

12) Set ATT/CMODE = $4000 (IDK mode), set ATT/DLU = ATT/BMODE = ATT/DT = 0, call RELFOR to release TRB and to release current format, call EXDFUN to release EOB program area; call CLRSR to clear exit stack and to exit to dispatcher.

INTERFACE

1. ENTRY METHOD

EOB is a disk resident program which is called via ENTPRG (Enter a disk resident program).

The first entry point is called by INTRGT when EOB is requested.

The second entry point is called by ERRKEY when batch validation error was overridden or by EOBIT at normal completion of EOB own-code.

2. INPUT PARAMETERS OR CONDITIONS

(I) = Address of the Active Terminal Table (ATT).

3. EXIT METHOD

Normal exit is via CLRSR (Clear exit subroutine), which exits to Dispatcher. Error exit is to ERROR if the function is illegal. Error exit is to TOTAL if a batch counter is nonzero. An exit is also made to EOBIT to execute EOB own code.

4. OUTPUT PARAMETERS OR CONDITIONS

(I) = Address of ATT.

ASSEMBLY CHARACTERISTICS

1. EQUATES

a. LOCORE

LPMASK ($2)
NZERO ($12)
ZERO ($22)
ONEBIT ($23)
ZROBIT ($33)
SST ($47)
CLCDWA ($90)
ENTSR ($91)
EXTSR ($92)

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME____EOB - End of Batch Processor_____Level 6

        CLRSR ($93)
        ERROR ($96)
        GETALP ($9B)
        ENTPRG ($9E)
        EXDFUN ($9F)
        DISPLY ($A0)
        SPILL ($A1)
        REWRIT ($A4)
        READP ($A5)
        REDDWA ($A9)
        WRTDWA ($AA)
        WRTATT ($AC)
        RELFOR ($AD)

b.   SST

        CLOCKS (17)
        STEMP (20)

c.   EST

        JDATE (40)

d.   ATT

        FTOB (1)
        CDMODE (2)
        JIDX (3)
        RCNT (4)
        ATRB (6)
        PFBA (7)
        CURDA (9)
        DT (13)
        BMODE (15)
        VRCC (16)
        BMDI (18)
        SW1 (19)
        SW2 (20)
        FUNIDX (21)
        ABTN (22)
        KEYDPR (26)
        VR (29)
        RCDER (30)
        RCD (32)
        DLU (35)
        BATCH (35)
        CTRS (36)
        ATEMP (73)
        TEMP5 (77)

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___EOB - End of Batch Processor___   Level 6

    e. EXTERNALS

        GFTLJD
        WSS
        EOBIT
        ADAYTO DAY (2 ASCII characters)
        AMONTO MONTH (2 ASCII characters)
        AYERTO YEAR (2 ASCII characters)
        CLRALL parameter for DISPLY to clear the screen
        SETDIS parameter for DISPLY to set screen location according to (A)
        FRZATT

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

PIO	CLCDWA	Calculate disk word address.
ENEXSR	CLRSR	Clear exits track and jump to dispatcher.
GSE	ERROR	Display terminal error.
SCANV	GETALP	Get an alpha character from input string.
ENTPRG	ENTPRG	Enter a disk resident program.
	EXDFUN	Release the disk resident program.
DISPLY		Clear display and set screen location to IDLE.
SPILL		Update the entry for the batch in the ABT.
	REWRIT	Rewrite current record.
	READP	Read record according to previous disk address saved in TRB.
	READN	Read next record.
	REDDWA	Read disk word address.
	FRZATT	Write ATT on disk (do not save the time).
GFCORE	RELFOR	Release format area or TRB.
GETLJD		Locate job in Legal Job Directory.
SKPDUP	EOBIT	Load and execute own code routine, then load and execute EOB.
	WSS	Write statistics record.

4. DATA AREAS USED

SST	System Status Table
EST	Extended System Table
ATT	Active Terminal Table
DTL	Data Track Layout
TRB	Terminal Record Buffer
SRL	Statistics Record Layout.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __EOB - End of Batch Processor__   Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $123_{16} = 291_{10}$ words

2. RELOCATION TYPE

    EOB is a disk resident program. It is written in run-anywhere code.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ EORENT - Write a Record to Disk _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    EORENT has two entry points.

    EORENT performs end of record functions: writes the current record to
    disk, updates counters in the Active Terminal Table, and so forth.

    FIRSTF locates the first field in a format group which requires data entry.

2. DETAILED DESCRIPTION

    a. EORENT

        1. (EORENT). Increment ATT/RCD by 1 (total number of records in batch).

            ● Call WRITEC to write the current record to disk.

            ● If current format has any verification fields, increment ATT/VRCD
              by 1 (number of verifiable records).

            ● If current record is flagged, increment ATT/RCDER by 1 (number of
              erroneous records).

            ● Move address of next record or disk from TRB/6/0-7 and TRB/5 to
              ATT/NXTDA and ATT/NEXDA + 1.

            ● Set ATT/SW1/14,15 = 0 (first record flags); and set ATT/SW1/2 = 1
              (first character of record is next).

            ● Set ATT/MODE/14 = 1 (IDK mode).

            ● If ATT/SW2/9 = 0 (no record insertion), reduce ATT/AVLTK
              (number of words left in current track) by the record length.

            ● If ATT/SW2/9 = 1 (record was inserted) set ATT/SW2/9 = 0.

        2. (NOSUB).

            ● Set ATT/RCNT = 0 (character count in current record) and
              ATT/FCNT/0-7 = 0 (character count in current field).

            ● Set ATT/RFA/9-15 = 0 (field number) and ATT/RFA/0-8 = 4
              (relative address in Format Table of current field information).

            ● Call FIRSTF to locate the first field in Format Table requiring
              data entry.

            ● Call DISPLY to display new header.

            ● If ATT/SW1/10 = 1 (record is not part of current Document group),
              go to step 4.

            ● Increment ATT/CRCD by 1 (complement of number records left in
              current document group). If not at end of current document group,
              go to step 4.

AA5641

- If ATT/ELDT/15 = 1 (end of document), go to step 3.

- Increment ATT/ELDT by 2 (relative address of current group in Legal Document Directory).

- Call ENTPRG to load and execute REDGRP, which reads in the next Format description from disk. Continue at step 4.

3. (ENDDOC). Call ENTPRG to load and execute REDDOC, which reads in the first Format description of the next document.

4. (ATTWRT). If ATT/RCD/0-3 ≠ 0 (number of records in batch is not a multiple of 16, exit to CLRSR.

- Call WRTATT to write ATT to disk, when number of records is a multiple of 16, then exit to CLRSR.

b. FIRSTF

1. (FIRSTF). Set ATT/ATEMP + 1 = 0 (Field number).

- If ATT/PFBA = 0 (no format defined), set ATT/FMTNO = 0 (Format number) and exit to caller.

- If autosequencing is specified for the format, increment ATT/ATEMP + 1 by 1 to allow for autosequencing field, and increment Q-register to bypass autosequencing field description.

2. (LOOP). If current field is not auto skip or auto dup field, go to step 3.

- If current field is auto dup, and ATT/SW1/15 = 1 (first record of format), go to step 3.

- If current field is auto dup, and ATT/SW1/12 = 1 (Inserted record) and FC bit (TRB/3/13) is on, go to step 3.

- If ATT/SW1/6 = 1 (see auto-fields are to be entered manually until next multi-level DUP/SKIP key) go to step 3.

- Otherwise, advance to next format in field, increment ATT/ATEMP + 1, and repeat step 2.

3. (FOUND). Call STTYPE to move data type from current format entry to ATT/DT/13-15.

- Exit to caller with (A) = Field number.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME_____ EORENT - Write a Record to Disk   Level 6

INTERFACE

1. ENTRY METHOD

    a. EORENT

        EXT   EORENT
        JMP   EORENT
        Exit is to CLRSR

    b. FIRSTF

        EXT FIRSTF
        RTJ FIRSTF
        --Return--

2. PARAMETERS

    (I) = Base address of Active Terminal Table (ATT)

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        CLRSR ($93)
        ENTPRG ($9E)
        DISPLY ($A0)
        WRITEC ($A3)
        WRTATT ($AC)

    b. SST

        CFFA (12)

    c. ATT

        MODE (2)
        FCNT (3)
        RCNT (4)
        ATRB (6)
        PFBA (7)
        AVLTK (8)
        NXTDA (11)
        FMTNO (13)
        RFA (14)
        ELDT (16)

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___EORENT - Write a Record to Disk_____Level 6___

                    SW1 (19)
                    SW2 (20)
                    CRCD (24)
                    VRCD (29)
                    RCD (32)
                    RCDSIZ (69)
                    ATEMP (73)

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    ENEXSR   (CLRSR)   Clear Exits stack and function lock
                ENTPRG   Load and execute a disk resident program
                DISPLY   Display a message at a terminal
    PI   WRITEC   Write current record to disk
                WRTATT   Write ATT to disk
    EMM   STTYPE   Move field data type to ATT

4. DATA AREAS USED

    SST   System Status Table
    ATT   Active Terminal Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $6E_{16} = 110_{10}$ words

2. RELOCATION TYPE

    Loader Relocatable.

3. ENTRANCY CLASSIFICATION

    EORENT   Reentrant
    FIRSTF   Reentrant

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    EORVR processes the end of record in verify mode. It also reads the
    next record, copies autodup fields as necessary, and performs basic
    checks to see if the record needs verification. If so, exit is to VMM1.

    The entry point EORVRC is used when the next record does not need to be
    read in, and entry point EORVR1 is used when erroreous record count
    should not be updated.

2. DETAILED DESCRIPTION

    (EORVR) Set ATT/ATTEMP + 2 to show no copying area currently in use.

    (NOVERY) Call REWRIT (PIO) if the record needs to be rewritten after a
    change was made. Check for an autodup change made in verify mode. If
    so, go to ADSVE. If not, check for an autodup change made in entry mode.
    If not, call READN to read the next record. Then jump to EORVRC. If
    so, set ATT/SW2/10 (autodup change flag). (ADSVE) Make a core request
    of CAM. Move the current record to the new core area. Read in the next
    record to the original current record core area and go EORVC.

    (EORVC) Check for same format as last record. If not, call GFCORE to
    get the new format release copying area if necessary, and go to SOF. If
    the autodup change was from entry, release copying area, and jump to SOF.
    If the autodup change was made in verify and the autodup change flag
    (ATT/SW2/10) is on, transfer all autodup field information into the next
    record. If autodup field to be exchanged has counter operations, subtract
    old value and add new value to accumulators concerned. Also update TRB
    Leader if any of the fields transferred are erroneous. (SOF1) Turn on
    rewrite flag (ATT/SW1/13). (SOF) Make basic checks to see if verification
    is needed on this format, for error flags, or for unbalanced counters.
    If so, set first record in format and clear autodup change flag if this
    is the first record. Call STTYPE (EMM) to set the data type in the ATT.
    Exit is to VMM1. If no verification is needed, write the checkpoint
    ATT if this is a sixteenth record jump to NOVERY.

INTERFACE

1. ENTRY METHOD

    Direct jump for all entry points.

2. INPUT PARAMETERS OR CONDITIONS

    1 = current ATT base address.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ EORVR - End of Record Processor - Verify Mode _____ Level 6

3. EXIT METHOD

    JMP   VMM1

4. OUTPUT PARAMETERS OR CONDITIONS

    1 = current ATT address

    The next record to be verified has been read in.

ASSEMBLY CHARACTERISTICS

1. a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($42)
        GFCORE ($9D)
        REWRIT ($A4)
        READN ($A6)
        WRTATT ($AC)
        CAM ($AE)
        WAIT ($B0)

    b. ATT

        RCNT (4)
        ATRB (6)
        PFBA (7)
        FMTNO (13)
        RFA (14)
        SW1 (19)
        SW2 (20)
        RCD (32)
        ATEMP (73)
        RCDER (30)

    c. SST

        CFFA (12)
        STEMP (20)

2. MACROS USED

    None

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___EORVR - End of Record Processor - Verify Mode_____ Level 6

3. PROGRAM DEPENDENCIES

VMM	(VMM1)	Exit to verify next record
EMM	(STTYPE)	Set data type information in ATT
	(COUNT)	Perform counter operations
PIO	(REWRIT)	Rewrite changed record on disk
	(READN)	Read next record from disk
	(WRTATT)	Write checkpoint ATT on disk
GFCORE		Get format in core
CAM		Allocate core
ENEXSR	(WAIT)	Delay
FMTPAR		Check for counter operations

4. DATA AREAS USED

ATT
SST (STEMP)
TRB

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$FB_{16} = 241_{10}$ words

2. RELOCATION TYPE

Loader relocatable.

3. ENTRANCY CLASSIFICATION

Dormant reentrant.

AA5641

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS _____ CYBERDATA

PROGRAM NAME _____ ERRKEY - Error Override Key   Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   This routine processes the error override key in all modes.
   Functions: Check for EOB error - if yes, exit. Otherwise, perform counter
   operations. Set error flag (field and record), clear IDK and ERR/OVR switches.

2. DETAILED DESCRIPTION

   a. If not IDK (ATT/2/14) and ERR/OR (SW2B12), exit to ERROR. Release func-
       tion (EXDFUN). Clear error flag (SW2B12) and check if EOB (SW1B0 and
       RCNT = 0). If so, clear EOB validation error (SW1B0) and exit to second
       entry point of EOB; otherwise, continue with step b.

   b. Gets first character of current field and set error bit (via GBYTE and
       SBYTE).

       Checks record flag (word 3 bit 15 in TRB). If not set, set it; else,
       go to step c.

       If ERRKEY is pressed because of replacing a field in Read mode positioned
       before the basic mode point, increase number of erroneous records
       (ATT, word 30).

   c. Check for counters on current field (FMTPAR), and perform if required
       (CONT). If not, RELEASE (SW1B0) continue with step d. Otherwise,
       clears ENT/EXT Stack, and exit to TSTEOR (entry point in SKPDUP).

   d. Clear IDK mode (ATT word 2 bit 14).
       If Verify or Pseudo Verify - Set SW1B13 (Rewrite record back on disk
       if correction was made), and exit to NOVAL (in VMM).
       If Entry or Pseudo Entry - Exit to VMM2.

INTERFACE

1. ENTRY METHOD

   ERRKEY is a disk-resident program which is invoked by ENTPRG.

2. INPUT PARAMETER

   (I) = Base address of Active Terminal Table (ATT).

3. EXIT METHOD

    a. To ERROR if ERR/OR is invalid at this point.

    b. To second entry point of EOB if during End of Batch.

    c. To TSTEOR if during RELEASE with AUTO ERROR FLAGGING.

    d. To NOVAL in Verify or Pseudo Verify.

    e. To EMM2 in Entry or Pseudo Entry.

4. OUTPUT PARAMETER

    (I) = Base address of Active Terminal Table (ATT).

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        ENTPRG ($9E)
        GTYPE ($94)
        SBYTE ($95)
        EXDFUN ($9F)
        ERROR ($96)
        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)

    b. ATT

        MODE (2)
        RCNT (4)
        SW1 (19)
        SW2 (20)
        ATRB (6)
        RCDER (30)
        CMODE (2)
        RCD (32)
        RCDRB (31)
        FCNT (3)
        ATEMP (23)

    c. CONSTANT

       A4 (4)
       STACKL (8)
       ENTR (79)

2. EXTERNALS

    COUNT
    EMM2
    NOVAL
    FMTPAR
    TSTEOR

3. PROGRAM DEPENDENCIES

          ENTPRG
    ENTPRG
          EXDFUN

    VMM   - NOVAL

          EMM2
    EMM
          COUNT

          GBYTE
    GSE   SBYTE
          ERROR

    FMTPAR

4. DATA AREAS USED

    ATT - Active Terminal Table.
    SST - System Status Table.
    TRB - Terminal Record Buffer.

DOCUMENT CLASS _____ CYBERDATA _____

PROGRAM NAME _____ ERRKEY - Error Override Key   Level 06 _____

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $60_{16} = 96_{10}$ words for program.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. REENTRANCY

    Reentrant.

DOCUMENT CLASS __ERRTAB__

PROGRAM NAME___Table of Cyberdata Error Codes___

GENERAL DESCRIPTION

****THE FOLLOWING TABLE CONTAINS 2-CHAR ERROR CODES TO BE DISPLAYED.
****ACCESS TO THE ENTRIES IS THE ERROR INDEX, WHICH IS THE DISTANCE
****TO THE REQUIRED ENTRY FROM THE ENTRY POINT, EPM.

*		*CODE*	*INDEX*	*MEANING*
EPM	ALF	1,	$0	CLEAR ERROR CODE
	ALF	1,A1	$1	DATA TYPE ERROR X
	ALF	1,A2	$2	SIGN MISSING
	ALF	1,A3	$3	ILLEGAL CHAR X
	ALF	1,A4	$4	ILLEGAL FUNCTION KEY
	ALF	1,A5	$5	MISMATCH CH X (Y)
	ALF	1,B1	$6	BEG OF DATA
	ALF	1,B2	$7	END OF DATA
	ALF	1,C1	$8	LIMIT ERROR
	ALF	1,C2	$9	SPECIAL TEST ERROR
	ALF	1,C3	$A	CHECK DIGIT ERROR
	ALF	1,C4	$B	OWN CODE TEST ERROR
	ALF	1,C5	$C	NON-ZERO ERROR
	ALF	1,D1	$D	INVALID OPERATOR NO
	ALF	1,D2	$E	INVALID FORMAT NO
	ALF	1,D3	$F	INVALID DOC NO
	ALF	1,D4	$10	INVALID JOB NAME
	ALF	1,D5	$11	INVALID BATCH NO
	ALF	1,E1	$12	ILLEGAL REQUEST
	ALF	1,E2	$13	ILLEGAL PARAMETER
	ALF	1,F1	$14	FORMAT NO IN DOC ERR
	ALF	1,F2	$15	OWN CODE ISNT ON DISK
	ALF	1,F3	$16	DISK FULL
	ALF	1,F4	$17	CORE FULL PLEASE WAIT
	ALF	1,F5	$18	FUNCTION BUSY PLEASE WAIT
	ALF	1,LL	$19	SYSTEM LOCK.WAIT
	ALF	1,PP	$1A	PARITY ERROR
	ALF	1,**	$1B	FIELD WITH ERROR FLAG
	ALF	1,C6	$1C	BALANCING ERROR
	ALF	1,D6	$1D	INVALID AUTO SEQ COUNT
	ALF	1,F6	$1E	DISK DESELECTED
	ALF	1,F7	$1F	BATCH FULL
	ALF	1,SM	$20	MESSAGE FROM SUPERVISOR

***INSERT ADDITIONAL ERROR CODES HERE
*
*

	EQU	EEPM(*)
	EQU	ERRTL (EEPM-EPM)
	END	

173

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   This routine processes the error override key in all modes.
   Functions: Check for EOB error - if yes, exit. Otherwise, perform
   counter operations. Set error flag (field and record), clear IDK and
   ERR/OVR Switches.

2. DETAILED DESCRIPTION

   a. If not IDK (ATT/2/14) and ERR/OR (SW2B12), exit to ERROR. Release
      function (EXDFUN). Clear error flag (SW2B12) and check if EOB
      (SW1B0 and RCNT = 0). If so, clear EOB validation error (SW1B0) and
      exit to second entry point of EOB; otherwise, continue with b.

   b. Gets first character of current field and set error (via GBYTE and
      SBYTE).

      Checks record flag (word 3 bit 15 in TRB). If not set, set it; else,
      go to c.

      If ERRKEY is pressed because of replacing a field in Read mode positioned
      before the basic mode point, increase number of erroneous records
      (ATT, word 30).

   c. Check for counters on current field (FMTPAR), and perform if required
      (CONT). If not, RELEASE (SW1B0) continue with d. Otherwise, clears
      ENT/EXT Stack, and exit to TSTEOR (entry point in SKPDUP).

   d. Clear IDK mode (ATT word 2 bit 14).

      If Verify or Pseudo Verify - Set SW1B13 (Rewrite record back on disk
      if correction was made), and exit to NOVAL (in VMM).
      If Entry or Pseudo Entry - Exit to VMM2.

2. INPUT PARAMETER

   (I) = Base address of Active Terminal Table (ATT).

3. EXIT METHOD

   1) To ERROR if ERR/OR is invalid at this point.

   2) To second entry point of EOB if during End of Batch.

   3) To TSTEOR if during RELEASE with AUTO ERROR FLAGGING.

   4) To NOVAL in Verify or Pseudo Verify.

   5) To EMM2 in Entry or Pseudo Entry.

4. OUTPUT PARAMETER

   (I) = Base address of Active Terminal Table (ATT).

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __ERRWEY - Error Override Key__   Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        ENTPRG ($9E)
        GTYPE ($94)
        SBYTE ($95)
        EXDFUN ($9F)
        ERROR ($96)
        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)

    b. ATT

        MODE (2)
        RCNT (4)
        SW1 (19)
        SW2 (20)
        ATRB (6)
        RCDER (30)
        CMODE (2)
        RCD (32)
        RCDRB (31)
        FCNT (3)
        ATEMP (23)

    c. CONSTANT

        A4 (4)
        STACKL (8)
        ENTR (79)

2. EXTERNALS

    COUNT
    EMM2
    NOVAL
    FMTPAR
    TSTEOR

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME ___ERRWEY - Error Override Key___   ___Level 6___

3. PROGRAM DEPENDENCIES

ENTPRG	ENTPRG
	EXDFUN
VMM	NOVAL
EMM	EMM2
	COUNT
GSE	GBYTE, SBYTE, ERROR
FMTPAR	

4. DATA AREAS USED

ATT - Active Terminal Table.
SST - System Status Table.
TRB - Terminal Record Buffer.

PHYSICAL CHARACTER

1. MEMORY SPACE REQUIREMENTS

$60_{16} = 96_{10}$ words for program.

2. RELOCATION TYPE

Disk resident, run anywhere.

3. REENTRANCY

Reentrant.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME ___ESB - End of Sub Batch Key___   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   This function checks EACH counter for non-ZERO value. Counters not equal
   to Zero are displayed, and in this case, the routine waits for continue
   or cancel.

   If cancel is received then before checking the next counter, the current
   one is set to Zero.

2. DETAILED DESCRIPTION

   a. First check the Patch mode. If not pure ENTRY, exit to ERROR. If
       not end of field to error; else check counter words. If all counters
       are Zero exit via CLRSR.

   b. If one counter word is not Zero, calculate the counter number, unpack
       the counter value and display: COUNT $n(1)$ = $n(2)\underline{+}$

       Where: $n(1)$ = Counter number
                $n(2)$ = Contents of this counter

       Then wait for CONT/CANC key to be pressed.
       If CANCEL, then clear the 4 counter words and continue to check
       the remaining counter words, as above.
       If CONTINUE depressed check remaining counters without clearing.

   c. After checking all the counters, if all were Zero exit via CLRSR;
       else, clear the screen and exit via CLRSR.

INTERFACE

1. ENTRY METHOD

   ESB is a disk resident function, called via the ENTPRG routine.

2. INPUT PARAMETERS

   (I) = Base address of Active Terminal Table (ATT)

3. EXIT METHOD

   Exit is to ERROR if an error is detected, to CLRSR otherwise.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____ESB - End of Sub Batch Key   Level 6_____

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        SST ($47)
        ERROR ($96)
        CLRSR ($93)
        DISPLY ($A0)
        ONEBIT ($23)

    b. ATT

        CMODE (2)
        FCNT (3)
        COUNT (36)
        ATTEMP (73)
        AINT (27).

    c. SST

        STEMP (20)

2. EXTERNALS

    CONCAN
    SETDIS
    DLINE
    CLRFLD
    REGEN
    UNP
    SETMOV
    GETBUF

3. PROGRAM DEPENDENCIES

    SETMOV
    DISPLY
    CONCAN
    CLRSR
    ERROR
    GETBUF

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___ESB - End of Sub Batch Key_____Level 6

4. DATA AREAS USED

ATT - Active Terminal Table.
Interrogate Buffer of 28 characters.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE

$76_{16} = 118_{10}$ for program.

28_{10} words for INTERROGATE BUFFER.

2. RELOCATION TYPE

Run anywhere.

3. REENTRANCY

Reentrant.

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME____ESH_____ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

The ESH (end-of-shift) command, is a supervisor function, used to
terminate CYBERDATA SYSTEM at the end of shift or day. It locks
all inactive terminals. If not all are inactive, it prints "terminals
still active" and the operators should suspend all activities before
the command is reentered.

2. DETAILED DESCRIPTION

Syntactic check of the command is done - SQZFLG indicates if SQZ was
required or not. (= 0, SQ2 should be done, = 3 should not be done).
In case of SYNTAX error exit is made to REJSUP.

The main loop runs all over the stations. For each station the current
working mode is checked. If it is active (supervisor mode, entry,
verify IdK, pseudo modes), a flag (FLAG) is set indicating there are still
active stations. If the station is in interrogate mode, CANCEL is inserted
in the TIQ. The program exist to WAIT for one cycle and the check is done
again for that station.

If the station is inactive, the screen is cleared, the message **END OF
SHIFT** is displayed, and the station is set to supervisor lock (error
lock and SW1 are cleared).

After all stations were checked it is checked if there were active stations.
If there were, the message "Terminals still active" is printed and the
program exits (to SU).

In case there was no active station, SQZFLG is checked. If SQZ was required,
the program releases itself (EXDFUN) and calls SQZ first overlay (via
ENTPRG). If SQZ was not specified, the message "All stations idle - perform
VX to terminate system" is printed.

The program releases itself and, after inserting EOR in interrogate buffer,
SOF supervisor function is called (via ENTPRG).

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME____ESH_____ Level 06 __

INTERFACE

1. ENTRY METHOD

    Supervisor function invoked by DSKSUP (via ENTPRG).

2. EXIT METHOD

    To SV   -   In case there are still active stations.

    To REJSUP   -   In case of command syntax errors.

    Via ENTPRG   -   In case of normal end of function.

3. OUTPUT PARAMETERS

    (A) = 2,3 in case of syntax errors (exit of REJSUP)

    (A) = 0
    (Q) = 80E   exit to ENTPRG (SQZ not required)

    (A) = 0
    (Q) = 3814   exit to ENTPRG (SQZ not required)

ASSEMBLY CHARACTERISTICS

1. EQUATES

    SST ($47)
    STEMP (20)
    SLOCK (1)
    ATTB (10)
    ATTLNG (19)
    CATT (11)
    IBL (14)
    TIQ (15)
    WAIT ($B0)
    GETCOM ($9C)
    ENTPRG ($9E)
    EXDFUN ($9F)
    DISPLY ($A0)
    SV ($B1)
    REJSUP ($B2)
    GETACP ($9B)
    LØCK (0)
    CMØDE (2)
    FUNIDX (21)
    AINT (27)

SW1 (19)
SCA (36)
FLAG (SCA + 10)
STANS (SCA + 11)
ATTSAV (SCA + 12)
SQZFLG (SCA + 13)
INDEX (SCA + 14)
TERNØ (SCA + 15)
TERATT (SCA + 16)
LPMASK (2)
ØNEBIT ($23)
ZRØBIT ($33)
ZERØ ($22)
M1 (71)
M2 (72)
M3 (73)

2. EXTERNALS

SETDIS
CLRALL
DLINE
PRTMSG
CANCEL

3. DATA AREAS USED

SST
ATT
TIQ
AINT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$9E_{16}$ = 158 words of program.
No buffer required.

2. RELOCATION TYPE

Loader relocatable.

3. ENTRANCY CLASSIFICATION

Not reentrant.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____ES1_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    ES1 is one of the overlays of the SQZ program which squeezes the LODD, FMTLNK, and the load program area (SIPRDN).

    ES1 deletes the inactive formats from the LFD (Legal Format Directory). Deletion is done only after checking that the inactive format is not needed. The check is done by searching all the data and verifying that the inactive format was not used in any record.

2. DETAILED DESCRIPTION

    ES1 checks that formats are really present in core, i.e. FLK (Formats Link Table) is not empty. Two buffers are allocated via CAM. One for the LFD, and the record for saving the inactive formats. If no core is available the program exits to REJSUP with error code.

    The LFD is read into core in sections (depending on the buffer lengths), and the inactive formats are stored in the second buffer. This step is finished when either all the LFD has been scanned or the inactive buffer is full.

    The SPILL tables are read one by one into their fixed core address (ABT listed in EST), and the current batches are read by the parts (BRGSEC internal routine) into the same buffer which serves for the LFD. The format number of each record and the significant format numbers in the ATT image in first track of batch are compared to the inactive formats listed in the inactive format's buffer. If any of them match an inactive format, the inactive format is flagged as useful (CHKINA internal routine) and a message is printed via SUPRW informing in which batch this format has been used.

    After all the present batches have been searched, the inactive formats which are not flagged are deleted from the LFD. Their entries are cleared by writing zeroes to the LFD on the system disk.

    If the inactive buffer filled up before all the LFD has been scanned, the process is done in more than one pass. The program has only one entry point - DLINA.

INTERFACE

1. ENTRY METHOD

    ES1 is an overlay of the SQZ program. It is loaded by ES4 overlay via ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

3. EXIT METHOD

    a. ES1 loads the next overlay of SQZ

        RTJ - (EXDFUN)   Release ES1
        CUR   A
        ENG   ES2
        RTJ   (ENTPRG)

    b. ENA   11   No memory available
        JMP   (REJSUP)

4. OUTPUT PARAMETERS

    a. I = current ATT base address.

    b. Word 46 in ATT is FLK current available entry.

    c. A message for each format that appears in the data

            FORMAT XXXX IS USED IN JOB jjjjjj , BATCH bbb

        Where: XXXX - format number
                JJJJJJ - job number
                   bbb - batch number

    d. In case there are no formats in the system ES1 prints via PRTMSG.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        ZERO ($22)
        ZROBIT ($33)
        ONEBIT ($23)
        LPMASK (2)
        NZERO ($12)

    b. TRANSFER VECTOR

        CLCDWA ($90)
        REDDWA ($A9)
        WRTDWA ($AA)
        REDSEC ($A7)
        WRTSEC ($A8)
        RELFOR ($AD)
        CAM ($AE)
        REJSUP ($B2)
        ENTPRG ($9E)
        EXDFUN ($9F)
        SUPRW ($97)
        ENTSR ($91), EXTSR ($92), RTNSR ($38)

   c. SST

      STEMP (20)
      STEMP1 (STEMP+1)
      SETMP2 (STEMP+2)
      STEMP3 (STEMP+3)
      SST ($47)

   d. EST

      LFD (39)
      MAKFMT (24)
      ABT (22)
      SPIL (23)

   e. ATT

      RCD (32)
      FMTNO (13)
      CMODE (2)
      BFMT (25)
      BMOD (15)
      DOCNO (17)
      DOCFMT (23)
      JOBIND (3)
      ATTEMP (73)

   f. TEMPORARIES IN ATT

      SCA (46)

Name	Address	Description
FLKLNG	(SCA)	– number of entries in FLK.
INABCK	(SCAT1)	– Backwards pointer of inactive buffer.
LFDBCK	(SCA+2)	– Backwards pointer of buffer used for LFD and for the data.
BUFLNG	(SCA+3)	– LFB buffer's length in words.
FMTNI	(SCA+4)	– Last format being checked for inactivity.
LFDPT	(SCA+5)	– Pointer to LFD buffer.
ALLINA	(SCA+6)	– Number of real inactive formats in buffer.
ABTPT	(SCA+7)	– Which ABT entry is checked.
CURTRK	(SCA+8)	– Which track of data is presently in core.
RECNO	(SCA+9)	– Number of records in the batch.
CURDIS	(SCA+10)	– The displacement in track of first word in data buffer.
ABTENT	(SCA+11)	– Number of active entries in ABT.
INATOP	(SCA+12)	– Pointer to the top of inactivities buffer.
TEMP	(SCA+13)	–

g. GENERAL EQUATES

SQZTRK ($80C)   - index of ES2 overlay.
SQZPRG ($80D)   - index of ES3 overlay.
FW (3) FWRITE
INALNG (100)   - inactives buffer length.

h. TRB

PRVLSB (1)
NSB (3)
NXTLSB (2)
FMTREC (4)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

	SETMOV	- Enable program moving.
	ENTSR	- Save return address in ENTR/EXIT stack.
ENEXSR	EXTSR	- Return via the ENTR/EXIT stack.
	RTNSR	- Remove last address from ENTR/EXIT stack and return it in Q.
	SUPRN	- Used to print informing message about inactive format being used.
	CLCDWA	- Calculate disk word address.
	REDDWA	- Read disk with word addressing.
PIO	WRTDWA	- Write disk with word addressing.
	REDSEC	- Read disk with sector addressing.
	WRTSEC	- Write disk with sector addressing.
GFCORE	RELFOR	- Release memory buffer.
CAM		- Allocate memory buffer.
	PRTMSG	- Print message according to parameter in A. (End of
SUPER		format squeeze message).
	REJSUP	- Print error message and exit.
ENTPRG	ENTPRG	- Load disk resident message and exit.
	EXDFUN	- Release disk resident function.
ARITH	BD	- Convert binary to decimal. Used to convert binary format number to decimal for informing message.
GETLJD		- Get job entry. Used to get the job name of the batch in which inactive format appeared for the informing message.

SYSTEM INSTALL PARAMETERS:

EXT SIABTS - ABT table length in sectors.
EXT SIFLNK - FLK disk address.

PRINTED IN THE USA

4. DATA AREAS USED

    ATT
    LFD
    FLK
    ABT
    DTL

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    Program length - $210_{16} = 528_{10}$.

    Inactive format's buffer - 103 words.

    LFD buffer - minimum - 93 words.

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Not reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME____ES2 - Second Overlay of SQZ._____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   ES2 squeezes the formats which appear in the LFD. All tracks in the
   FLK which are not full are released after the formats in these tracks
   have been transferred to other tracks which were allocated by DAM and
   listed in the FLK. The new disk address for the transferred formats
   are updated in the LFD.

2. DETAILED DESCRIPTION

   Three memory buffers are allocated via CAM. One, FRMTRK, 51 words long
   (16 3+3 hdr words) for the list of formats which resides in specific
   track. Each format needs a 3 word entry: its number, its sector address
   in track and its length. The maximum number of formats in a track is 16.
   The second buffer serves for the LFD,LFDBCK, and the third one is for
   the formats FMTBCK. Their length depends on the memory available. If
   not enough available, the program exits to REJSUP.

   The FLK is read entry by entry. Each track address is checked against
   the LFD which is read by parts to the LFD buffer. Each format resides
   in this track entered to the FMTTRK. The process is finished when it
   is verified that the formats residing in this track fill it completely,
   or when all the LFD has been scanned. If a track is full it is listed as
   an entry in the FLK. If the track is not full, its formats are transferred
   to another track which was allocated by DAM. The formats transferred to
   new tracks are read into the memory buffer FMTBCK (by parts if necessary),
   and then written to the new track. As soon as a format is transferred
   to the new track, its sector address in the LFD is updated.

   To avoid loss of information in case of power fail, when a new track is
   allocated, it is written to the entry after the last in the FLK. When
   the first new track is allocated, the header of the FLK is updated in
   the following way: number of available sectors on current track = 0,
   current entry is increased by one.

   When a new track has not enough memory for a format which must be
   transferred, it is listed as an entry in the FLK. When all formats of
   a track have been transferred, it is released and its entry is cleared.

   When all the tracks listed in the FLK have been checked, the process is
   finished. The last target track is written to the FLK and the FLK
   header is updated with number of available sector on the last target
   track and the FLK entry on which it has been written.

   ES2 has one entry point FMTSQ.

   If DAM does not succeed to allocate the first new track, the process is
   supressed and the program exits to REJSUP with reject message code. If
   the process finishes successfully a message 'FORMAT TABLE SQUEEZED' is
   printed via PRTMSG.

PAGE NO. ___307___

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME ___ES2 - Second Overlay of SQZ.___   Level 6

INTERFACE

1. ENTRY METHOD

    ES2 is an overlay of SQZ loaded via ENTPRG by ES1.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.
    Word 46 in ATT is the current entry in FLK.

3. EXIT METHOD

    a. Normal Exit:

        RTJ - (EXDFUN)
        CLR   A
        ENQ   ES3
        RTJ - (ENTPRG)

    b. Error Exit:

        1) ENA   11   NO MEMORY AVAILABLE
            JMP-   (REJSUP)

        2) ENA   51   DISK IS FULL
            JMP-   (REJSUP)

4. OUTPUT PARAMETERS OR CONDITIONS

    a. I = current ATT base address.

    b. If program completes successfully the message:

            FORMAT TABLE SQUEEZED is printed.

    c. Error code 51 (disk is full) for REJSUP.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        ZERO ($22)
        ZROBIT ($33)
        ONEBIT ($23)
        NZERO ($12)

b. TRANSFER VECTOR

    CLCDWA ($90)
    REDDWA ($A9)
    WRTDWA ($AA)
    REDSEC ($A7)
    WRTSEC ($A8)
    RELFOR ($AD)
    CAM ($AE)
    REJSUP ($B2)
    ENTPRG ($9E)
    EXDFUN ($9F)
    ENTSR ($91)
    EXTSR ($92)
    RTNSR ($98)

c. SST

    SST ($47)
    STEMP (20)

d. EST

    LFD (39)
    MAXFMT (24)

e. ATT

    LUN (35)

f. TEMPORARIES IN ATT

    SCA (46)
    FLKLNG (SCA)   - number of entries in ATT.
    FMTBFL (SCA+1)   - length of formats buffer.
    FMTTRK (SCA+2)   - backwards pointer of formats of track buffer.
    FMTBCK (SCA+3)   - backwards pointer of formats buffer.
    LFDBCK (SCA+4)   - backwards pointer of LFD buffer.
    LFDPT (SCA+5)   - pointer to LFD buffer.
    CURFLK (SCA+6)   - entry in FLK whose track is checked against LFD.
    NEWFLK (SCA+7)   - entry in FLK on which the next full track is listed.
    RENSEC (SCA+8)   - free sectors on new track.
    NEWTRK (SCA+9)   - new track disk address.
    CURTRK (SCA+S)   - disk address of the track which is checked
                         against the LFD.
    FMTNO (SCA+11)   - last format whose LFD entry was checked.
    FLK   (SCA+12)   - FLK disk address.
    BUFLNK (SCA+13)   - length in words of LFD buffer.

    g. GENERAL EQUATES

       PRGSQ ($30D)   - next overlay function index for ENTPRG.

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

PIO	CLCDWA	-	calculate disk word address
	WRTDWA	-	write disk with word addressing
	REDDWA	-	read disk with word addressing
	REDSEC	-	read disk with sector addressing
	WRTSEC	-	write disk with sector addressing
ENEXSR	ENTSR	-	save return address in Enter/Exit track
	EXTSR	-	return via the Enter/Exit track
	RTNSR	-	remove last address from Enter/Exit track
	SETMOV	-	enable program moving
GFCORE	RELFOR	-	release memory buffer.
CAM		-	allocate memory buffer.
ENTPRG	ENTPRG	-	load disk resident program.
	EXDFUN	-	release disk resident function.
DAM	GATRT	-	get one track.
	RELTRT	-	release track.
SUPER	REJSUP	-	print error message and exit.
	PRTHSG	-	print error message.

System install parameters:

EKT   SIFLNK - FLK disk address.

4. DATA AREAS USED

    ATT
    LFO
    FLK
    FMTLNK

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___ES2 - Second Overlay of SQZ._____Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY REQUIREMENTS

    249_{10} = 99 + 99 + 51- minimum requirement for memory buffer.

    188_{16} = 382_{10} - program size.

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Not reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___ ES3 - Third overlay of SQZ - Program Squeeze   Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    ES3 is one of the overlays of the SQZ program which squeezes up the LOD,
    FMTLNK and the load program area (SIPRDN).

    ES3 responsibility is to remove the unoccupied holes in the Load Program
    Area.

    The PRD (program directory) is searched, and programs in the Load Program
    Area are compressed if possible, and their address in the PRD on disk and
    in memory is updated accordingly. In order to "avoid" destruction of a pro-
    gram, the program is first transferred to its place in the Load Program
    Area. It will be transferred back to programs area in the next pass of SQZ.

2. DETAILED DESCRIPTION

    ES3 allocates a memory buffer via CAM for programs transferring. The current
    available disk address in the Load Program Area is initialized to the start
    address of this area (SIPRDN). The PRD is searched. If a program was left
    in the work area (SIDSQZ) in the previous pass, it is transferred back to its
    original disk area (via TRAN subroutine) according to the address which was
    left in the third word of program's header in the previous pass of ES3.

    Disk addresses of programs are taken from the PRD or from the program header
    if the program is at present in core. If program whose disk address equals
    the current available disk address is found, the current available disk
    address is increased by the program length in sectors. If no program whose
    disk address is equal or greater than the current available disk address and
    less than the next available load address is found, no more transferring
    should be done. Otherwise the program closest to the current available
    address is transferred. First it is transferred (via TRAN routine) to the
    fixed work area on disk (SIDSQZ) with its original disk address written in
    the program header, and the PRD entry on disk is updated with this address.

    Only then is the program transferred to the current available address, and
    the disk address in PRD on disk is updated. The address in PRD in core or
    in program header (if presently in core) are also updated. Current available
    address is increased by program length in sectors. If no more transferring
    can be done, the next available disk address for loading programs in PRD is
    updated and the program exits.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___ES3 - Third overlay of SQZ - Program Squeeze_____Level 06___

INTERFACE

1. ENTRY METHOD

    ES3 is an overlay of the SQZ program. It is loaded by ES2 overlay via
    ENTPRG and executed from its only entry point PRGSQ.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

3. EXIT METHOD

    a. ES3 releases its area via EXDFUN and loads ESH via ENTPRG. ESH is
        executed from the second entry point.

        RTJ-   (EXDFUN)   release ES3

        ENA   1   second ENTRY POINT

        LDQ   =N$3819   ESH

        RTJ-   (ENTPRG)

    b. ENA   11

        JMP- (REJSUP)

4. OUTPUT PARAMETERS OR CONDITIONS

    a. I = current ATT base address.

    b. The following message is printed via PRTMSG PROGRAM LIBRARY SQUEEZED.

ASSEMBLY CHARACTERISTICS

1. EQUATES

a. ZROBIT ($33)
    ONEBIT ($23)
    ZERO ($22)
    NZERO ($12)
    LPMASK (2)

b. SST

SST	($47)
STEMP	(20)
PRO	(3)

c. EST

OWCNO	(4)
PRDDA	(10)

d. TRANSFER VECTOR

CAM	($AE)
REJSUP	($B2)
RELFOR	($AD)
CLCDWA	($90)
WRTDWA	($AA)
REDSEC	($A7)
WRTSEC	($A8)
ENTSR	($31)
EKTSR	($32)
RTNSR	($98)
ENTPRG	($9E)
EXDFUN	($9F)
WAIT	($BO)

e. ATT

ATTEMP	(73)	
SCA	(46)	
IND	(SCA)	TRAN ROUTINE PARAMETER
BUFBCK	(SCA+1)	BACKWARDS POINTER FOR CORE BUFFER
BUFSLN	(SCA+2)	LENGTH (IN SECTORS) OF CORE BUFFER
CURAD	(SCA+3)	DISK ADDRESS TO WHICH WE MOVE
MAXPRD	(SCA+4)	NUMBER OF ENTRIES IN PRD
NXTLOD	(SCA+5)	NEXT AVAILABLE DISK ADDRESS FOR LOAD PROGRAM
MINDIS	(SCA+6)	CURRENT MINIMAL MISTAKE TO CURRENT ADDRESS

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ ES3 - Third overlay of SQZ - Program Squeeze   Level 06___

MINFUN	(SCA+7)	ADDRESS IN PRD OF PROGRAM WITH MINDIS
MINBIT	(SCA+8)	CONTROL BIT OF PROGRAM WITH CURRENT MINIMAL DISTANCE
PRGLEN	(SCA+9)	PROGRAM LENGTH
PRGBIT	(SCA+10)	CONTROL BIT OF PROGRAM

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   ENEXSR
   - SETMOV
   - ENTSR
   - EXTSR
   - RTNSR
   - WAIT

   PIO
   - CLCDWA
   - WRTDWA
   - REDSEC
   - WRTSEC

   ENTPRG
   - ENTPRG
   - EXDFUN

   SUPER
   - PRTMSG
   - REJSUP

   GFCORE
   - RELFOR

   SYSTEM INSTALL PARAMETERS

        SIDSQZ - working area for transferring programs.

        SIPRDN - beginning of "load program" area.

4. DATA AREAS USED

   ATT

   PRD

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___ES3 - Third overlay of SQZ - Program Squeeze_____Level 06_____

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$125_{16} = 293_{10}$ words for the program.

99 words - minimum size of memory buffer.

2. RELOCATION TYPE

Run anywhere.

3. ENTRANCY CLASSIFICATION

Not reentrant.

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    ES4 is the first overlay of SQUEEZE program. The documents directory
    (LDD) is searched and the documents table (LDT) is compressed so unused
    portions are eliminated. ES4 also clears 4 first control words in Tape
    Recovery Area.

2. DETAILED DESCRIPTION

    The point of compression is initiaized as the entry right after the LDD.

    Two buffers are allocated: one for the LDD and the other is used for the LDT.
    The LDT buffer serves two purposes: One half serves as target buffer
    (compressed LDT) and the other as source buffer (original LDT). Two temporaries
    in the ATT. CURBG1, CURBG2 are the corresponding displacements in LDD and LDT
    of these two buffers.

    The LDD is read into core and searched. If a document whose displacement
    equals the current point of compression is found, the point of compression
    is increased with the document length and the process continues for the
    new point of compression. If a document whose displacement is greater
    than point of compression is not found, compressing had been finished. The
    current available displacement in LDD + LDT (in LDD header) is updated. The
    LDD is written to disk (via WRTLDD) subroutine) and the target buffer,
    provided it is not empty, is written to the LDT (via WRTMVE subroutine). The
    four control words of tape recovery area are cleared, buffers are released
    and the program exits.

    If documents whose displacements are after the current available displacement
    are found, the document closest to the point of compression is transferred
    from the source buffer to the target buffer and its displacement in LDD
    core buffer is updated.

    During transfer, portions of the document should be read into source buffer
    (via RDMVE subroutine) if not available at core, and the target buffer will
    be written to the LDT (via WRTMVE subroutine) every time it is filled up.
    If any writing is done while transferring, the LDD and the target buffer are
    written to disk. The current point of compression is updated and the process
    continues.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___ES4 – First Overlay of SQUEEZE_____ Level 06

INTERFACE

1. ENTRY METHOD

    ES4 is first of SQZ overlays, loaded by ESH via ENTPRG.

        CLR   A
        LDQ   =N$80E
        RTJ - (ENTPRG)

2. INPUT PARAMETERS OR CONDITIONS

    All stations are supervisor locked.
    I = current ATT base address.

3. EXIT METHOD

    a. ES4 loads next overlay ES1 after releasing itself.
        RTJ - (EXDFUN)
        CLR   A
        LDQ   =XFMTSQZ   ES1
        RTJ - (ENTPRG)

    b. If no memory is available for the two buffers, the program exits to
        REJSUP with the appropriate error code:

        ENA   11
        JMP - (REJSUP)

4. OUTPUT PARAMETERS OR CONDITIONS

    a. Before the normal exit, two informing messages are printed via PRTMSG.

                        'DOCUMENT TABLE SQUEEZE'
                        'TAPE RECOVERY AREA CLEARED'

    b. When the program exits via REJSUP, error code 11 is output parameter.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        ZROBIT ($33)
        ONEBIT ($23)
        ZERO ($22)

    b. SST

        SST ($47)
        STEMP (20)

    c. EST

        LDD (3)

    d. TRANSFER VECTOR

        CAM ($AE)
        REJSUP ($B2)
        RELFOR ($AD)
        CLCDWA ($90)
        REDDWA ($A9)
        WRTDWA ($AA)
        REDSEC ($A7)
        WRTSEC ($A8)
        ENTSR ($31)
        EXTSR ($92)
        RTNSR ($98)
        ENTPRG ($9E)
        EXDFUN ($9F)

    e. ATT

        ATEMP (73)
        SCA (46)
        LDDBCK (SCA+1)   - backwards pointer of LDD buffer.
        MVBFL (SCA+3)   - length of LDT working buffer (in words).
        MVBCK (SCA+4)   - backwards pointer of LDT working buffer.
        CURBG1 (SCA+5)   - displacement of LDD+LDT of the beginning of target buffer.
        CURBG2 (SCA+6)   - displacement of LDD+LDT of the beginning of source buffer.
        MINDIS (SCA+7)   - displacement of the document currently closest to point of
                           compression.

```
        DOCNO   (SCA+8)  - document number.
        MINDOC  (SCA+9)  - document currently closest to point of compression.
        DOCLEN  (SCA+10) - the length of MINDOC.
        NEWMVE  (SCA+11) - the displacement of the free word in target buffer.
        CURMVE  (SCA+12) - the displacement in source buffer of the word to be
                           transferred.
        CURDIS  (SCA+13) - current point of compression (displacement in LDD+LDT).
        FLGDD   (SCA+14) - flag set if writing has been done during transferring.
```

f. GENERAL EQUATES

    FMTSQZ ($808)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

```
                ENTSR
                EXTSR
        ENEXSR  RTNSR
                SETMOV


                CLCDWA
                REDDWA
        P10     WRTDWA
                REDSEC
                WRTSEC


                ENTPRG
        ENTPRG  EXDFUN


        GFCORE  RELFOR


        SUPER   REJSUP


                PRTMSG


        CAM     CAM
```

4. DATA AREAS USED

    ATT
    LDD + LDT
    TRA - Tape Recovery Area

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME __ES4 - First Overlay of SQUEEZE__ _____ Level 06

Table on Disk
LDD+LDT

LDT
working buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $148_{16} = 328_{10}$ program length.

    $512_{10} + 3 =$ LDD memory buffer.

    $80+3 =$ LDT double purposes buffer minimum length.

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Not reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __EXPTAB__   Level 6

PROGRAM FUNCTION

EXPTAB resides in disk and takes the form of a miscellaneous program. It is used by the XPL supervisor command and is composed of 2 tables:

1. A string of ADC's which define the lengths of the layout texts of the Supervisor commands.

2. The layout texts themselves where each text is divided into display lines and the length in words of each line preceding the line itself.

Access to the table is through entry 21 in the miscellaneous section of the PRD.

When the table is accessed only the first table, or one command layout text or the first line of any of the layout texts is called into core.

SIZE

$583_{16} = 1411_{10}$ words.

CONTROL DATA CORPORATION ⊛ PSSD ⊛ SOFTWARE DOCUMENT

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___FAD - Format Adjustment Based on Keyed Data_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

FAD is a disk resident program, classified as an end of field own-code test. When specified in the field definition in the format, it is called via ENTPRG in order to change the format according to the data field, but not at beginning of record. The data field is used as a key for a table look-up. The table is user written (i.e., the user must reassemble the program in order to insert the required table). Each entry in the table is composed of a key whose length is equated in the program by the user (up to 8 words), followed by the format number hexadecimal.

2. DETAILED DESCRIPTION

FAD first enters SETMOV to enable program moving and then checks ATT/ Current mode for entry or insert (ATT/SW1/12); if neither exit is to ERROR to indicate own code error. Sets look-up key length according to field length and up to the length equated in the program. Check data type in ATT and sets parameters to collect look-up key characters. (When data type is ALPHA starts at left-most character of the field, when numeric starts at right-most character). When the key has been formed, a loop-through the table is begun.

When the key is found in the table, the format number associated with that key is extracted and saved; if the key is not found exit is to ERROR. According to ATT/RFA+2, a core request is issued and the required portion of the old format is saved. The new format is called via CFCORE and if not found the requested core is released and exit is to ERROR (EREXT2). If the format is inactive the old format is recalled (EREXT1), and control is passed to EREXT2.

If the format is in order, a loop is performed to check the corresponding fields in both formats to ascertain that they are equal in length and data type. If a discrepancy is found, goes to EREXT2.

The record defined by the new format is not longer than the record defined by the old format goes to update TRB header according to the new format (LNGOK). But if the record is longer and it is pure entry record after insert, it will be written in the address specified as "next disk address" in the TRB header. Then it is checked if there is enough room for this record on the current track; if so goes to LNGOK, else a new track is requested via G1TRT and linked (via LINKTR) and the "disk address of next record is updated in the header of the previous record. Then the ATT/RFA is set according to the new format, "1st record in format" bit (ATT/SW1/15) is set, and TRB header is updated. The previously requested core is released via·RELFOR, the program releases itself via EXDFUN and returns to caller via EXTSR.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____FAD - Format Adjustment Based on Keyed Data_____ Level 6

INTERFACE

1. ENTRY METHOD

    FAD is a disk resident own-code routine loaded by ENTPRG. The only
    entry point is FAD.

2. INPUT PARAMETERS OR CONDITIONS

    I = ATT address.

3. EXIT METHOD

    a. Jump to error with own code error code.

    b. The program releases itself via EXDFUN
        Jump to EXTSR.

4. OUTPUT PARAMETERS OR CONDITIONS

    I = ATT address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        SST ($47)
        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)

    b. SST

        STEMP (20)

    c. TRANSFER VECTOR

        CLCDWA ($90)
        ENTSR ($91)
        EXTSR ($92)
        GTBYTE ($94)
        ERROR($96)
        CAM ($AE)
        GFCORE ($9D)
        EXDFUN ($9F)
        REDDWA ($A9)
        WRTDWA ($AA)
        RELFOR ($AD)

    d. ATT

        CMODE (2)
        FCNT (3)
        RCNT (4)
        ATRB (6)
        PFBA (7)
        AVLTK (8)
        NDA (11)
        DT (13)
        FMTNO (13)
        RFA (14)
        SW1 (19)
        SW2 (20)
        LUN (35)
        ATEMP (73)

    e. CONSTANTS

        OCERR ($B)
        ERRF3 ($16)
        CORFUL ($17)
        KEYLNG (according to user requirements)

2. PROGRAM DEPENDENCIES

FCORE	-	-	Call a format and link it to the ATT.
	(RELFOR)	-	Release to format or allocatable core.
ENEXSR	- (ENTSR)	-	Save return address in the common enter/exit stack.
	- (SETMOV)	-	Enable moving of disk resident program.
	- (EXTSR)	-	Common subroutine exit.
GSE	- (GBYTE)	-	Extract one character from TRB.
	- (ERROR)	-	Handle error conditions.
CAM	-	-	Allocate core buffer.
ENTPRG	- (EXDFUN	-	Release disk resident program.
PIO	- (CLCDWA)	-	Calculate disk record address.
	- (REDDWA)	-	Read disk according to word addressing.
	- (WRTDWA)	-	Write disk according to word addressing.

3. DATA AREAS USED

    Old format buffer
    New format buffer

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __FAD - Format Adjustment Based on Keyed Data__   Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $16E_{16} = 366_{10}$ words + length of user written table.

2. RELOCATION TYPE

    Run anywhere.

3. REENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME__ FIELDS - Display a Field   Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    FIELDS has three entry points:

    CFLD displays a field.
    CWFLD displays a field number and the field.
    CSTVER displays a field for sight verification.

2. DETAILED DESCRIPTION

    a. CFLD

        1) o If not Read mode go to a2.

            o Get first character of field if error flagged go to a3.

        2)   Store given parameters in TOPT/11,12 and go to a4.

        3) o Increase cursor address by 1 and convert to to output buffer
               via CONCSR.

            o Prepare parameters and call CDATA to display ** for error
               flagged field.

        4) o Increase cursor address by 2 and construct the first word
               of the parameters list of CDATA call.

            o If not under Pseudo Entry go to a5.

            o If it is not the field just entered go to a5.

            o Adjust the number of characters to display according to field
               count (ATT/FCNT).

        5)   Call CDATA to display the field and exit to caller via EXDSR.

    b. CWFLD

        ● Call ENDSR to save return address in Exits stack.

        o Advance screen address to beginning of next line.

        ● Call CFDNOS to display field number at beginning of next line.

        ● Call FMTSCN to extract field parameters. Use field address and
           field length in calling CFLD to display field.

        ● Exit to caller via EXDSR.

c. CSTVER

   1) (CSTVER). Call ENDSR to save return address in Exits stack.

      ● If ATT/CMODE/12 = 1 (Pseudo Verify mode), go to step 2.

      ● Backspace current screen position by 2.

      ● Call FMTSCN to extract field parameters. Use field address and field length in calling CFLD to display field.

   2) (PVER). Set cursor position to one column ahead of end of field.

      ● Call CONCSR to convert cursor position to line number and position within line.

      ● Call CDATA to display cursor.

      ● Exit to caller via EXDSR.

INTERFACE

1. ENTRY METHOD

   EXT A   where A = CFLD, CWFLD, or CSTVER
   RTJ A
   --Return--

2. INPUT PARAMETERS OR CONDITIONS

   (I) = Base address of Active Terminal Table (ATT)
   For CFLD: (ATT/RQ/15) = Byte number
              (ATT/RQ/0-6) = Character count
              (ATT/RQ + 1) = Address of field
   For CWFLD: (ATT/IOADDR) = Field number

3. EXIT METHOD

   Return to location immediately following RTJ instruction.

4. OUTPUT PARAMETERS OR CONDITIONS

   (I) = Base address of ATT.

PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME__ FIELDS - Display a Field   Level 06

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        CDATA ($8D)
        ENDSR ($8E)
        EXDSR ($8F)

    b. ATT

        CMODE (2)
        MODE (2)
        FCNT (3)
        ATRB (6)
        FLDNO (14)
        TOPT (33)
        IOADDR (64)
        RQ (65)

    c. TOPT

        CURSRD (3)
        PARAS (8)
        FLDST (15)

    d. CONSTANT

        C (1) Temporary storage constant index

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    VL2CRT   (CDATA - Display driver initiator)
              (CARLIN - Characters per line)
              (CONCSR - Convert to screen address)

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME __FIELDS - Display of Fields__ __Level 06__

        HDRNOS   (CFDNO - Display field number)

        FMTSCN   Extract field parameters

        CMISC1   (ENDSR - Display entry routine)
                    (EXDSR - Display exit routine)

4. DATA AREAS USED

    ATT   Active Terminal Table
    TOPT Terminal Output Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMFNTS

    $81_{16} = 129_{10}$ words.

2. RELOCATION TYPE

    Core resident, loader relocatable.

3. ENTRANCY CLASSIFICATION

    Passive reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___FMSEL - Format Select_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    This routine processes the Format Select key and the keys following.
    It performs the following operations:

    a. Checks that format select is legal, and if not, exits to error.

    b. Calls GETSTR to read in up to 4 characters + RELEASE. On return,
        if cancel key was depressed, clears up and exits.

    c. Calls GETBIN to convert characters received. If nonnumeric or
        format number greater than maximum set by system, exits to error.

    d. Calls GFCORE. On return if inactive, releases and exits to error.
        Else, display format, clear data line where Fxxx appeared previously,
        set SW1's bit 10 (see ATT), and exit via EXDFUN and CLRSR.

2. DETAILED DESCRIPTION

    a. FMSEL checks if SW1 bit no. 2 is set (see ATT word no. 19). If not,
        control is transferred to ERROR.

    b. Calls DISPLY to clear field number, and then stores F (in ASCII)
        in Current Char (see ATT - CURCH), and invokes DISPLY again for
        displaying the character F.

    c. Sets bit no. 15 of DT/FMTNO (see ATT word no. 13), and displays
        a J. Then invokes GETSTR which initiates a read of 5 characters
        from the key station into the interrogate buffer, and returns to
        FMSEL with completion code (see INT word no. 2). FMSEL checks the
        completion code.

    d. If cancel received, calls DISPLY to clear first data line of display
        and replace it by the string: '001/', sets up first field to be entered
        parameters (FIRSTF), and exits via CLRSR.

    e. If release key was not received, FMSEL checks the address of entry in
        the FBA table which contains the address of the current format (see
        ATT word no. 7).

    f. If zero, sets SW1 bit no. 3 (see ATT word no. 19) and SW1B10 if in
        Insert, and then transfers control to ERROR. Else, calls RELFOR to
        release FBA, clears the word (ATT word no. 7) and places zero in the
        Format Number (see ATT word number 13). Sets SW1 bit no. 3 (see ATT
        word no. 19) and goes to f.

g. If release received, calls GETBIN, which converts the string to binary number, and control is returned to FMSEL. If the format number is illegal (not numeric, or equal to zero), then processing continues at e. Else, stores the number in the first temporary working area. (see SST word no. 20).

Checks if format is larger than maximum format number allowed by the system (see EST word no. 24). If yes, then proceeds as described in e. If not, invokes GFCORE which releases the previous format, looks for required format in core, and if not found, on disk.

If the format does not exist, or is too long for the current job, processing continues at e.

h. Checks if the format is active (see FMT word no. 3 bit no. 15) or inactive.

i. If active, then if Insert Flag (SW1 bit 12) is set, then resets SW1 bit no. 3 and sets Format Change bit (see TRB word no. 3 bit no. 13), and calls DISPLY to replace the first data line by the string '001/' and exits as in d.

If Insert flag is not set, then sets SW1 bit no. 10 and 15, clears bit no. 3, calls DISPLY to replace the first data line by the string; '001/', and continues as in d.

j. If the format is inactive, then calls RELFOR which decrements the user count (see FMT word no. 2) and releases core if necessary.

k. Clears FBA address (see ATT word no. 7), and also clears the format number (ATT word no. 13).

l. Sets SW1's bit no. 3 and control is transferred to e.

INTERFACE

1. ENTRY METHOD

    FMSEL is a disk resident program which is called via ENTPRG routine.

2. INPUT PARAMETERS

    Register I which contains the base address of Active Terminal Table (ATT).

3. EXIT METHOD

    Two exits from FMSEL:

    a. If an error is detected the exit is via GSE (Entry Point ERROR).

    b. Exit via CLRSR.

4. OUTPUT PARAMETERS

    a. I = base address of ATT.

    b. Q = error index, when ERROR is invoked (see ERRTAB- core resident routine).

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        DISPLY ($A0)
        GFCORE ($9D)
        GETSTR ($A2)
        GETBIN ($99)
        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZEROBIT ($33)
        ERROR ($96)
        RELFOR ($AD)
        SST ($47)
        CLRSR ($93)

    b. ATT

        CMODE (2)
        AINT (27)
        CURCH (2)
        SW1 (19)
        PFBA (7)
        FDTP (13)

    c. INDEX

        ERRA4 (4)
        ERRD2 ($E)

2. PROGRAM DEPENDENCIES

    a. ROUTINES CALLED

        ENEXSR: SETMOV - Set Move
        GSE: ERROR
        DISPLY: DISPLAY
        PIO: GETSTR - Get String of Char.
        SCANV: GETBIN - Get the Number in Binary from Input String
        GFCORE GFCORE - Get Format Into Core
               RELFOR - Release Format
        CLRSR: Clear Exit Subroutine

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___FMSEL - Format Select_____Level 6_

    b. EXTERNAL LIST

       FIRSTF
       DFMT
       SETMOV
       DCHAR
       CLRFLD
       DFNDTY
       DDOC
       DBOF

3. SYSTEM TABLES USED

    SST - System Status Table (see Cyberdata software user's guide)
    ATT - Active Termianl Table (see Cyberdata software user's guide)
    EST - Extended System Table (see Cyberdata software user's guide)
    FMT - Format Table (see Cyberdata software user's guide)
    FBA - Format Base Address Table (see Cyberdata software user's guide)
    FRP - Format/Records/Programs Buffer (see Cyberdata software user's guide)
    INT - Interrogate Input Buffer (see Cyberdata software user's guide)

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $6F_{16} = 111_{10}$ words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. REENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___ FMTADV - Format Advance _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    FMTADV proceeds the advance to the next group in document. Legal only
    under Document at beginning of record in ENTRY mode.

2. DETAILED DESCRIPTION

    a. FMTADV exits via ERROR if one of the following occurs:

        1) Record is under F. SELECT.

        2) Mode is not ENTRY.

        3) Not at beginning of record.

    b. Invokes EXDFUN which decrements the user count by one (see FRP word
        no. 3), updates the time when the program area will be free, and returns
        control to FMTADV.

    c. Checks if the current group is the last group in document (see ATT word
        no. 16, bit no. 15).

        If not, then it updates the disk displacement of the current group of
        the document from the beginning of the LDD (see ELDT in ATT), and
        invokes ENTPRG which brings in the routine REDOC, and transfers control
        to this routine for executing from the entry point: REDGRP.
        Else (The last group in document was reached), calls ENTPRG which
        transfers the control to REDOC.

INTERFACE

1. ENTRY METHOD

    EMTADV is a disk resident program which is invoked by ENTPRG.

2. INPUT PARAMETERS

    a. (I) = base address of Active Terminal Table (ATT).

    b. SWI bit no. 10 (ATT word no. 19) = 1 when current record is part of the
        document or F. SELECT. (Set to one by F. Select, ENT and REDOC when
        document no. is illegal).

    c. CMODE (ATT word no. 2 bit 11-8) = current mode for operator:

        0000 = IDLE
        0001 = VERIFY
        0010 = READ
        0011 = ENTRY

d. RCNT (ATT word no. 4) = character count in current record.

e.. End of document (ATT word no. 16 bit no. 15) = This bit is set to
one when the last group in the document is reached.

3. EXIT METHOD

Exit is to ERROR if an error is detected (see Detailed Description a.),
else, exit is to ENTPRG.

4. OUTPUT PARAMETERS

1. EQUATES

a. LOCORE

LPMASK (2)
NZERO ($12)
ONEBIT ($23)
ZROBIT ($33)
ERROR ($96)
EXDFUN ($9F)
ENTPRG ($9E)

b. ATT

SW1 (19)
CMODE (2)
RCNT (4)
ELDT (16)

c. ERROR CODE

CA4 ($4)

2. PROGRAM DEPENDENCIES

GSE: ERROR

ENTPRG $\begin{cases} \text{EXDFUN - EXIT DISK FUNCTION} \\ \text{ENTPRG - ENTER PROGRAM} \end{cases}$

3. DATA AREAS AND SYSTEM TABLES USED

a. SST - see Cyberdata Software User's Guide..

b. FRP - see Cyberdata Software User's Guide.

c. ATT - see Cyberdata Software User's Guide.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___FMTADV - Format Advance___ Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $1C_{16} = 28_{10}$ words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. REENTRANCY CLASSIFICATION

    Dormant reentrant.

AA5641

PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____FMTBAK - Format Backward_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   FMTBAK backs up the previous entry in the document. If it already
   was the first entry exits to ERROR.

2. DETAILED DESCRIPTION

   a. FMTBAK exits via ERROR if one of the following occurs:

      1) Record is under F. SELECT.

      2) Mode is not ENTRY.

      3) Not at beginning of record.

   b. Reads the entry in LDD from the disk. If, while working on the
      document, the supervisor has deleted it, calls ERROR (F1).

   c. If it is the first entry in the document - exits to ERROR (A4).
      Else, replace the pointer to entry in LDT by the pointer to
      entry in LDT of the previous group of the document.
      (See ELDT - ATT word no. 16:   INA - 2
                                      STA - ELDT,I).

   d. Invokes ENTPRG to load and execute REDGRP.

INTERFACE

1. ENTRY METHOD

   FMTBAK is disk resident. It is called via ENTPRG routine.

2. INPUT PARAMETERS

   a. I = base address of Active Terminal Table (ATT).

   b. SW1 bit no. 10 = current record is part of the document or F. SELECT.
      Set to zero by DOC and RTD.
      Set to one by F. SELECT, ENT and REDOC when document no. is illegal.

   c. ELDT (ATT word no. 16) = disk displacement of the current group of the
      document from the beginning of the LDD.

   d. CMODE (ATT word no. 2 bit 11 - 8) = current mode for an operator.

      0000 = IDLE
      0001 = VERIFY
      0010 = READ
      0011 = ENTRY

   e. RCNT (ATT word no. 4) = character count in current record.

   f. LDDDA (SST word no. 9) = LDD disk address.

   g. DOCNO (ATT word no. 17 bit 15-8) = document number.

3. EXIT METHOD

Exit is to ERROR if an error is detected (see 2a, 2b, 2c).
Else, exit is to REDGRP via ENTPRG.

4. OUTPUT PARAMETERS

a. a = Error code when ERROR is invoked.

b. I = Base address of ATT.

c. SW1 bit 3,10 (ATT word no. 19) = see Input Parameters - b, and ATT.

d. ELDT = See Input Parameters - c.

ASSEMBLY CHARACTERISTICS

1. EQUATES

a. LOCORE

SST ($47)
ERROR ($96)
CLCDWA ($90)
REDWA ($A9)
EXDFUN ($9F)
ENTPRG ($9E)
LPMASK (2)
NZERO ($12)
ZERO ($22)
ONEBIT ($23)
ZROBIT ($33)

b. ATT

SW1 (19)
CMODE (2)
RCNT (4)
DOCNO (17)
ATEMP (73)
ELDT (16)

c. SST

EXTSST (0)

d. EST

LDDDA (9)

e. ERROR CODES

CA4 ($4)
CF1 ($14)

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME____FMTBAK - Format Backward_____Level 6

2. PROGRAM DEPENDENCIES

PIO
$\begin{cases} \text{CLCDWA - Calculate Disk Word Address} \\ \text{REDWA \ - Read Disk Word Address} \end{cases}$

GSE   - ERROR

ENTPRG
$\begin{cases} \text{EXDFUN - Exit Disk Function} \\ \text{ENTPRG - Enter Program} \end{cases}$

3. DATA AREAS AND SYSTEM TABLES USED

   a. LDD See Cyberdata software user's guide.

   b. LDT See Cyberdata software user's guide.

   c. SST See Cyberdata software user's guide.

   d. ATT See Cyberdata software user's guide.

   e. GST See Cyberdata software user's guide.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$3D_{16} = 61_{10}$ words.

2. RELOCATION TYPE

Disk resident, run anywhere.

3. REENTRANCY CLASSIFICATION

Reentrant.

DOCUMENT CLASS ___FMTPAR_____

PROGRAM NAME___Locate a designated parameter in Format Table_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    FMTPAR has two entry points: FMTPAR and GFNAME. FMTPAR extracts a
    designated format parameter from the current format definition.

    GFNAME locates the name of the current field in the format definition
    table.

2. DETAILED DESCRIPTION

    a. FMTPAR

        1. Set SST/STEMP + 2 = 0 (counter).
           Set SST/STEMP + 3 = address of word 2 of current field description
           in Format Table (FMT). Mark out the flag bit addressed by Q, and
           save it in SST/STEMP + 1.

        2. (CHECK). Count the number of parameters that precede the designated
           parameter in the field description. Keep the count in SST/STEMP + 2.

        3. (OUT). If designated parameter is not defined for this field,
           set (A) = $FFFF and exit to caller.

        4. (FOUND). Pick up designated field value in A/0-7 with A/8-15 = 0.
           Set (6) = Address of word containing field, and exit to caller.

    b. GFNAME

        1. Call FMTSCN to locate field description in Format table. Extract
           name length from field description. If name length = 0, exit
           to caller with (A) = 0.

        2. (GFN2). Call FMTPAR to locate limit parameter within field
           description. If limit parameter exists, go to step d. If no
           limit parameter exists, calculate address of name parameter using
           parameter count returned in Q.

        3. (GFFXIT). Exit with (A) = name length and (Q) = name address.

        4. (LIMIT). Calculate address of name parameter using address of
           limits parameters and number of words in upper and lower limits.
           Go to step 3.

INTERFACE

1. CALLING SEQUENCE

    a. FMTPAR

        EXT   FMTPAR
        ENQ   B   B = Position of flag bit in word 2
                                                      of field description

DOCUMENT CLASS ___FMTPAR___

PROGRAM NAME___Locate a designated parameter in Format Table___

```
RTJ         FMTPAR
  -- Return--
(A) = $FFFF if requested parameter        B = 15 Resequencing
      is not defined.                          14 Fill Character
    = value of requested parameter             13 Own Code
      if it is defined.                        12 Special Test
                                               11 Counters
(Q) = count of defined parameters              10 Limits
      which precede the requested
      parameter (count starts with
      bit 15), if required parameter
      not found.

    = address of parameter if defined.
```

b. GFNAME

```
EXT         GFNAME
ENA         N                    N = Field Number
RTJ         GFNAME
  --Return--
(A) = 0 if name is not defined
    = number of words in name, if defined

(Q) = address of first word of name, if defined
```

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)

    b. SST

        CFFA (12)
        STEMP (20)

    c. FMT

        LIMBIT (10)

    d. FMTSCN

        FN1 (5)

2. MACROS USED

    None

DOCUMENT CLASS __FMTPAR__

PROGRAM NAME___Locate a designated parameter in Format Table___

3. PROGRAM DEPENDENCIES

FMTSCN   Locate field description in Format Table

4. DATA AREAS USED

SST   System Status Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$45_{16} = 69_{10}$ words

2. RELOCATION TYPE

FNTPAR is a core-resident, loader relocatable program.

3. ENTRANCY CLASSIFICATION

Dormant Reentrant

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___FMTSCN___

PROGRAM NAME___Locate Field Description in Format Table___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   FMTSCN locates a field description in the current format table (FMT),
   based on either field number or column number input. Information about
   the field (field number, column number, field length, etc.) is returned
   in a parameter list within FMTSCN.

2. DETAILED DESCRIPTION

   a. (FMTSCN). Set counters (column counter CN1 and field counter FN1)
      to maximum positive value ($7FFF). Set column number (CN) and field
      number (FN) to zero. Set relative address of field description (FMTRFA)
      to 4 (relative address of first field description in format). Set
      address of field in Terminal Record Buffer (TRBFAD) to address of first
      field in TRB. Set (Q) = Address of first field in format description.
      If field number specified, store field number in FN1. If column
      number specified, store it in CN1.

   b. (NXTFLD). Store current field length in FLDLNG. Decrement column
      number (CN1) by current field length. When CN1 becomes negative
      (we have reached the requested field), go to step d. Decrement field
      number (FN1) by 1. When FN1 becomes negative (we have reached the
      requested field), go to step d. If current field is the last field
      in the format, go to step c. If not, update Q and FMTRFA by number
      of words in current field, update CN by number of characters in
      current field, increment FN by 1, and repeat step b.

   c. (LSTFLD). Set (Q) = $FFFE and exit to caller.

   d. (LOOPEX). Save address of current field definition in FN1. Save
      address of current field in TRB in TRBFAD. Set FLDLNG/15 = 0 if
      current field begins in bits 8-15 of word, = 1 if current field
      begins in bits 0-7 of word.

      Set (Q) = Address of parameter list beginning at FN, and exit to
      caller.

INTERFACE

1. ENTRY METHOD

   EXT   FMTSCN
   RTJ   FMTSCN

2. INPUT PARAMETERS OR CONDITIONS

   (A) = Field Number
       = Column Number, with bit 15 = 1

   (I) = Base Address of Active Terminal Table (ATT).

DOCUMENT CLASS ___FMTSCN___

PROGRAM NAME___ Locate Field Description in Format Table___

3. EXIT METHOD

Return to location immediately following RTJ FMTSCN.

4. OUTPUT PARAMETERS OR CONDITIONS

(A) = Undefined

(Q) = $FFFE if requested field not located in Format Table

= Address of Parameter List, if field was located,

Word	0	Field Number
	1	Column Number
	2	Relative Address of Field Description in Format Table (FMT)
	3	Address of Field in Terminal Record Buffer (TRB)
	4	Length of field (Bit 15 = 1 if first character of field is in bits 0-7 of word)
	5	Address of Field Description in FMT

(I) = Base Address of ATT

ASSEMBLY CHARACTERISTICS

1. EQUATES

a. LOCORE

LPMASK (2)
ZERO ($22)

b. ATT

TRB (6)
FBA (7)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

None

4. DATA AREAS USED

ATT   Active Terminal Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$45_{16} = 69_{10}$ words

163

DOCUMENT CLASS ___FMTSCN___

PROGRAM NAME___Locate Field Description in Format Table___

2. RELOCATION TYPE

FMTSCN is a core resident, loader relocatable program.

3. ENTRANCY CLASSIFICATION

Not applicable

164

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___FRZ - Freeze Batches_____ Level 6

PROGRAM DESCRIPTION

1. GENERAL DESCRIPTION

    This is a supervisor function whose duty is to assign active batches the
    frozen state, and the stations they were connected to, the idle state. A
    check-point corresponding to the time the batch was frozen, with certain
    flags and disk addresses is written on the ATT image in order to allow
    this state to be restored.

2. DETAILED DESCRIPTION

    The supervisor issues the command:

            FRZ,n(1) [,n(2)]

    where:

            n(1),n(2) are station numbers
            n(2) is optional

    If n(2) is not specified it is considered n(2) = n(1).

    The station numbers are checked as follows:

            $n(2) \geq n(1)$
            $n(2) \leq$ number of stations which were initialized at system start-up

    The ATT's of stations n(1) to n(2) are checked. Stations in supervisor or
    idle mode are skipped. If the station is in ENTRY or VERIFY mode the following
    checks are made:

    If it is under function or pseudo function lock the
    program enters the WAIT state until the locks are cleared.

    If the station is reading through an interrogate buffer the CANCEL code is
    put into the TIQ to cause a cancel of the request, and the execution of the
    program is delayed.

    The function lock is set on the supervisor's station in order to stop new
    characters from entering the TIQ. The supervisor's ATT base address and
    its function index are saved in the station ATT, and the I register is
    assigned the station's ATT base address. The executing station is now the
    "frozen" station.

The interrogate and search buffer are released. Their backwards
pointers will be used for saving necessary disk addresses. The
"override monitor checks" but is set to enable unusual modifications
of the ATT. If the batch is in End of Data state the most significant
bit in the AINT (interrogate backwards pointer) is set. CURDA is
saved in AINT and ASRCH. If the rewrite bit is set, the current
record is rewritten. The rewrite bit is set again in case rewriting
will be needed. If the station is in ENTRY mode or INSERT mode and
the record in the TRB is not empty, it is written to disk through
WRITEC. When the batch is thawed the last processed record is read
according to CURDA, and the CURDA is updated with the values saved
in AINT and ASRCH.

If the batch is in ENTRY mode and under document control, the ELDT
(displacement in LDT) is changed to relative mode (displacement from
the beginning of the document). The accumulated time of the batch
is updated. The TRB is released and if it is End of Field, $FF is
saved in FCNT. The format is released and the PFBA is used to save
the error lock. RCNT is transferred to complement form (-RCN) and
the ATT is written on the ATT image in the first track of batch.
The negative RCNT will identify a check point that was written by FRZ
from other checkpoints.

The batch status in the ABT is changed to "frozen". The station is
assinged the Idle mode, its screen is cleared and then the message
"BATCH FROZEN" is printed on it. Several words in the ATT of the
station, and the locks, are cleared.

When all the stations that were specified in the command are frozen
a message with all the station numbers which were in ENTRY or VERIFY
mode is printed on the initiating device.

INTERFACE

1. ENTRY METHOD

    The program is loaded via DSKSUP after the FRZ command has been issued
    in the supervisory mode.

2. ENTRY PARAMETERS

    The FRZ command is in the interrogate buffer.

3. EXIT METHOD

Normal: The program exits through SV after all the stations which were
specified in the command are frozen.

Error: If station numbers in the command are illegal exit is done
through REJSUP.

4. OUTPUT PARAMETERS OR CONDITIONS

Normal exit: a) The message BATCH FROZEN is displayed on the screen
of every station where a batch was frozen.

b) The message
FRZ HAS BEEN DONE ON THE FOLLOWING STATIONS:

n(1), n(2), ... n(m)

where n(i) is the number of a station whose batch was frozen is printed
on the supervisor's station.

Error exit: The code of invalid terminal number is in A register.

ASSEMBLY CHARACTERISTICS

1. EQUATES

a. LOCORE

UPMASK (2)
NZERO ($12)
ZERO ($22)
ONEBIT ($23)
ZROBIT ($33)
TEN ($46)

b. SST

SST ($47)
SLOCK (1)
STEMP (20)
ATTLNG (13)
BATT (10)
CFFA (12)
CATT (11)
IBL (14)
TIQ (15)
CUOCKS (17)

c. EST

LDDA (9)

d. ATT

    NDA (11)
    CMODE (2)
    CURDA (9)
    RCNT (4)
    RCD (32)
    DT (13)
    FCNT (3)
    ATEMP (73)
    AINT (27)
    FUNIDK (21)
    ASRCH (28)
    SW1 (19)
    SW2 (20)
    BMOD (15)
    DOCNO (17)
    ELDT (16)
    LUN (35)
    FTOB (1)
    TRB (6)
    PFBAC (7)
    JIDK (3)
    ABTIDX (22)
    SC (36+10)
    FTER (SC)   First Station Number in the Command
    LTER (SCH)   Last Station Number in the Command
    FLAG (SC+2)
    ASCII (SC+3)

e. TRANSFER VECTOR

    GETALP ($9B)
    GETBIN ($99)
    REJSUP ($B2)
    SV ($B1)
    WAIT ($B0)
    RELFOR ($AD)
    REWRIT ($A4)
    WRITEC ($A3)
    WRTATT ($AC)
    SPILL ($A1)
    DISP ($A0)
    ENTSR ($91)
    EXTSR ($92)
    CLCDWA ($90)
    REDDWA ($A9)
    SUPRW ($97)

f. GENERAL EQUATES

FW(3)

2. MACROS USED

None.

3. PROGRAM DEPENDENCIES, ROUTINES USED

ENEKSR
$\left\{\begin{array}{l}\text{ENTSR}\\ \text{EXTSR}\\ \text{SETMOV}\\ \text{WAIT}\end{array}\right.$

SUPER
$\left\{\begin{array}{l}\text{REJSUP}\\ \text{SV}\\ \text{PRTMSG}\end{array}\right.$

PIO
$\left\{\begin{array}{l}\text{REWRIT}\\ \text{WRITEC}\\ \text{FRZATT}\\ \text{CLCDWA}\\ \text{REDDWA}\\ \text{WRTDWA}\\ \text{GETBUF}\\ \text{MOVREC}\end{array}\right.$

DISP
$\left\{\begin{array}{l}\text{SETDIS}\\ \text{REGEN}\\ \text{DLINE}\end{array}\right.$

CANCEL

4. AREAS USED

SST
ATT
TIQ
TRB
DTL
ATT image

AA5641

PRINTED IN THE USA

DOCUMENT CLASS **CYBERDATA IMS**

PROGRAM NAME **FRZ - Freeze Batches**   Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY REQUIREMENTS

    Program length $15D_{16} = 349_{10}$.

2. RESIDENT TYPE

    Disk resident.

3. REENTRANCY

    Reentrant

4. RELOCATION TYPE

    Run anywhere, moving.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME_____ GETLJD - Manipulate an Entry in the Legal Job Directory _____ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    GETLJD is the core resident portion of a program to enter, locate, delete, or modify an entry in the Legal Job Directory (LJD). The program is locked against additional callers while the disk resident portion (GJB) is loaded and executed.

2. DETAILED DESCRIPTION

    a. (GETLJD). If (FLOCK) ≠ 0 (GETLJD locked)

        Call RETADD to compute caller's retry address.
        Call STRTRY to schedule retry address.

    b. (OK). If GETLJD not locked, lock it. (set FLOCK = $FFFF).

    c. Save parameters:

        (Q) in FLAG
        Job Index in ASTORE
        Address of caller's parameter list in LOC

    d. Call ENTSR to save parameter list address in exit stack.
       Save ATT/FUNIDX (caller's function index) in SAVIDX.
       Call RETADD to calculate length of caller's parameter list.

    e. (LOOP). Move caller's parameter list to TAB.

    f. (SAVADD). Call ENTPRG to load and execute GJB. On return, restore ATT/FUNIDX to caller's function index. Call RTNSR to retrieve caller's parameter list address from exit stack. Call RETADD to calculate length of caller's parameter list; using that value, calculate return address.

    g. (LPZ). Move parameter list from TAB to caller's parameter list.

    h. (OUT). Unlock GJD (set FLOCK = 0) and exit to caller.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___GETLJD - Manipulate an Entry in the Legal Job Directory_____Level 06

INTERFACE

1. ENTRY METHOD

    GETLJD is a core resident routine.
    The calling sequence depends on the requested function:

        ENQ   Function Code
        RTJ   GETLJD
        ADC ⎤
         ●   ⎬ Parameters
         ●   ⎪
        ADC ⎦
        --Retry Return--
        --Normal Return--

    The parameter list varies with the function code:

Code	Function	Parameter List
0	Enter new job	5-word LJD entry
1	Locate job	3-word job name and 2 words for returned information
2	Locate job	5 words for returned information
3	Delete job	None
4	Modify job	2 words containing LJD entry words 3 and 4

2. EXIT METHOD

    Exit is to the Retry Return if the GETLJD program is locked. Otherwise, exit is to the Normal Return.

3. OUTPUT PARAMETERS OR CONDITIONS

(Q) = $FFFF	Job does not exist and LJD is full.
(A) = undefined	
(Q) = 0	Job does not exist. If (q) = 0 on entry, new job is entered.
(A) = undefined	
(Q) = 1	Job does exist.
(a) = Job index	If (Q) = 1 on entry, LJD entry words 3 and 4 are returned in parameter list. If (Q) = 2 on entry, LJD entry words 0 to 4 are returned in parameter list.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK ($2)
        SST ($47)
        ENTSR ($91)
        RTNSR ($98)
        ENTPRG ($9E)
        STRTRY ($AF)

    b. SST

        STEMP (20)

    c. ATT

        JIDX (3)
        FUNIDX (21)
        ATEMP (73)

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    ENTSR   Save return address in exit stack.
    RTNSR   Return exit address from exit stack.
    ENTPRG Enter and disk resident program.
    STRTRY Schedule retry address.

4. DATA AREAS USED

    SST   System Status Table
    ATT   Active Terminal Table

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___GETLJD - Manipulate an Entry in the Legal Job Directory_____Level 06

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $4C_{16} = 76_{10}$ words.

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Not reentrant.

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___GETPL - Get Page and Line of Field___ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    GETPL finds the screen addresses of the requested and next format fields.
    If the requested field is the last one in the format, the page of the
    next field is set to minus zero. Otherwise, the caller is supplied with
    a pointer to a block of four words containing:

    a. Page number of requested field.

    b. Screen address within page of requested field.

    c. Page number of next consecutive field (FFFF if no next consecutive
       field).

    d. Screen address within page of next consecutive field.

2. DETAILED DESCRIPTION

    GETPL saves the requested field number in REQFLD (SST + TEMP + 2) and sets
    a pointer to the beginning of the format. It initializes the output block
    with the current field number as zero and the screen address as the beginning
    of the data area on the screen.

    GPLO calculates the number of screen locations required for the current field
    plus 2 locations for a possible error flag plus the number of locations for
    the field name at the beginning of the line (NAMFLD). This number is divided
    by the number of characters per line (CARLIN). The quotient is increased by 1
    and multiplied by the number of characters per line. This result is the number
    of characters required for the current field. A running total is kept in LOCN2
    and this value is checked to see if a new page is needed for the next field.
    If so, go to NEWPAG.

    GPLO1 checks the current field against the requested field. If the current
    field is greater than the requested field, then go to GPLEX because the caller's
    request has been completed. Otherwise, go to GPLO2.

    GPLO2 moves the address for the next consecutive field into the address for
    the requested field to prepare for the calculation of the next field address.
    A check is made to see if the current field is the last field. If so, PAGE2
    is set to FFFF and control goes to GPLEX. If not, go to TONEXT to increment
    the current field number, readjust the format pointer to the new field, and
    jump to GPLO.

    NEWPAG set LOCN2 to the beginning of data area on the screen (LOW + 1), moves
    the current field number to PAGE2. (The page number is expressed as page
    beginning with field "n", not as page 1, page 2, etc.) If the current field
    number is greater than the required field number, go to GPLEX. Otherwise go
    to GPLO.

DOCUMENT CLASS CYBERDATA

PROGRAM NAME GETPL - Get Page and Line of Field   Level 06

3. PROGRAM DEPENDENCIES

VL2CRT contains CARLIN
                 LOWDAT
                 UPDAT
                 NAMFLD

4. DATA AREAS USED

SST
ATT
Internal storage of parameter block and pointers

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$46_{16} = 70_{10}$ words.

2. RELOCATION TYPE

Loader relocatable.

3. ENTRANCY CLASSIFICATION

Not reentrant.

DOCUMENT CLASS ___GFCORE___

PROGRAM NAME___Get Format/Release Format___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    GFCORE has two entry points: GFCORE and RELFOR. GFCORE ensures that a designated format description is present in core, by locating the format in core or by reading the format from disk, and establishes linkage to the current Active Terminal Table (ATT).

    RELFOR releases a format in core, if there are no other users involved.

2. DETAILED DESCRIPTION

    a. GFCORE

    1. Call ENTSR to save the return address in the exit stack. Save the requested format number in ATT/ATEMP. If Format no. requested = 0, call MONITOR (Error 13).

    2. (NOTZERO). If requested format number = ATT/FMTNO/0-12 (current format number), set (A) = $FFFF and call EXTSR to exit. Otherwise, set ATT/SW2/3 = 1 (monitor ATT check override). If ATT/PFBA = 0 (no current format), go to step 3. Otherwise, call RELFMT to release current format, and set ATT/PFBA = ATT/FMTNO = 0.

    3. (ON). Save requested format number in SST/STEMP. Search formats in core for format number which matches the requested format number. If no match, go to step 5.

    4. (FOUND). If a matching format number is found in memory, increment the user count in the format header by 1, save the address of the format block in ATT/ATEMP + 3, save the corresponding Format Base Address (FBA) table address in ATT/PFBA, set (Q) = address of format and go to step 9.

    5. (DSKCHK). Call CLCDWA to calculate disk word address of requested format. Call REDDWA to read a 2-word entry from the Legal Format Directory (LFD) into ATT/ATEMP + 1 and ATT/ATEMP + 2.

    6. (CORERQ). If ATT/ATEMP + 1 = 0 (Format does not exist), set (A) = 0 and go to step 11. If the format does exist, call CAM to allocate a format buffer. If space is not available, call WAIT to wait one cycle, then repeat step 6.

    7. (COREFD). Save the format buffer address in ATT/ATEMP + 3. Set the I/O bit in the format header, and call REDSEC to read the format from disk to the format buffer area.

84

8. (START). Put the format buffer address in the first available entry in the FBA. Put the FBA entry address in ATT/PFBA and in word 0 (Backwards pointer) of the format header. Clear the I/O bit in the format header.

9. (STOREF). Put format number in ATT/FMTNO. Compute the current format field address, and save it in SST/CFFA. If current terminal is a Supervisor, call EXTSR to exit. If it is not a Supervisor terminal, call STTYPE to put the current format data type in ATT/DT/13-15. Compare the maximum record length for the format with the Terminal Record Buffer (TRB) length. If there is insufficient space in the TRB, set (A) = 0 and go to step 11.

10. (ENDBIG). Call DISPLY to display the format number. Clear ATT/ SW2/3 (monitor ATT check override), set (A) = $FFFF and call EXTSR to exit.

11. (EREXIT). If ATT/ATEMP/15 = 1 (Return to caller) clear ATT/SW2/3 (monitor ATT check override), set (A) = 0, and call EXTSR to exit.

12. (NORTRN). If format was read into core, call RELFOR to release format.

13. (DSPERR). Set ATT/SW1/3 = 1 (DOC, FORMAT ADV, or F. SELECT must be keyed in next), set ATT/FMTNO/0-12 = 0 (current format number), call DISPLY to blank out display number, clear ATT/SW2/3 (monitor ATT check override), set (Q) = ERRD2 and exit to ERROR.

b. RELFOR

1. Decrement user count in format header by one. If user count ≠ 0, exit to caller.

2. Set Format Base Address (FBA) table entry = 0. Set SST/1/14 (Storage Move Flag) = 1, and exit to caller.

INTERFACE

1. CALLING SEQUENCE

a. GFCORE

```
EQU        GFCORE ($9D)
ENQ        F                        B/0-14 = Format Number
RTJ        (GFCORE)                 B/15 = 0, exit to ERROR in care of error
--Return--
```

(A) 0 if format is in core

                        (A) = 0, if error occurred

(Format does not exist, or insufficient space in TRB)

DOCUMENT CLASS ___GFCORE_____

PROGRAM NAME_____Get Format/Release Format_____

    b. RELFOR

        EQU   RELFOR ($AD)
        LDG   Format Address
        RTJ-   (RELFOR)
           --Return--

In each case (I) = Base address of Active Terminal Table

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        MONITR ($8C)
        CLCDWA ($90)
        ENTSR ($91)
        EXTSR ($92)
        ERROR ($96)
        DISPLY ($A0)
        REDSEC ($A7)
        REDDWA ($A9)
        RELFMT ($AD)
        CAM ($AE)
        WAIT ($B0)

    b. SST

        CFFA (12)
        STEMP (20)
        LFD (39)

    c. EST

        FBA

    d. ATT

        ATRB (6)
        PFBA (7)
        FMTNO (13)
        RFA (14)
        SW1 (19)
        SW2 (20)
        ATEMP (73)

86

DOCUMENT CLASS _____

PROGRAM NAME_____

e. ERROR CODE

   ERRO2 ($E)

2. MACROS USED

   None

3. PROGRAM DEPENDENCIES

MONITR	Checkout monitor routine
CLCDWA	Calculate disk word address
ENTSR	Save return address in exit stack
EXTSR	Exit to last return address in exit stack
ERROR	Error procedure
DISPLY	Display message
REDSEC	Read a sector
REDDWA	Read disk word address
CAM	Core allocation module
WAIT	Delay return for one cycle
STTYPE	Move data type from format description to ATT
ABSADD	Ensure correct addressing

4. DATA AREAS USED

SST	System Status Table
EST	Extended System Table
ATT	Active Terminal Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $B7_{16} = 183_{10}$ words

2. RELOCATION TYPE

   GFCORE is core-resident, loader-relocatable.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS __GJB__

PROGRAM NAME__ Maintain Legal Job Directory__

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    GJB is the disk resident portion of the Legal Job Directory (LJD) main-
    tenance function. It is called by GETLJD to perform one of 5 possible
    maintenance functions:

    Enter a new job in LJD.
    Given the job name, locate a job in LJD.
    Given the job index, locate a job in LJD.
    Delete a designated job from LJD.
    Modify 4th and 5th words of the designated LJD entry.

2. DETAILED DESCRIPTION

    a. (GJB). Call SETMOV to enable GJB to be moved, if necessary.

        o Call a procedure to perform the requested function:

            Function Code = 0, Enter New Job   (Step b)
                          = 1, Locate Job by Job Name   (Step d)
                          = 2, Locate Job by Job Index   (Step e)
                          = 3, Delete Job   (Step f)
                          = 4, Modify Job   (Step g)

    b. (NEWJOB). Call GETJOB to locate an available entry in the Legal Job
        Directory (LJD). If the designated name already exists in LJD, or
        if there are no available entries in LJD, go to step h with (Q) = 1
        or $FFFF.

        o If a LJD entry is available, call ADDR to calculate the disk
           word address of the available entry.

        o Call DWRITE to write the new entry in LJD.

        o Increment the job count by one (via ADDR and DWRITE).

    c. (WJBNO). Call ADDR to calculate the disk word address of the job
        count in LJD.

        o Call DWRITE to write the new job count in LSD.

        o Go to step h with (Q) = 0.

    d. (RDNAME). Call GETJOB to locate entry in LJD for designated Job Name.

        o If the job does not exist in LJD, go to step h with (Q) = 1 pr $FFFF.

        o Move the 4th and 5th word from LJD entry to core resident parameter
           list.

        o Go to step h with (Q) = 1.

49

e. (RDIDX). Call ADDR to calculate the disk word address of the designated job in the LJD.

- Call REDDWA to read the LJD entry for the designated job.

- If the LJD entry is empty, go to step h with (Q) = 0.

- If not, go to step h with (Q) = 1.

f. (DELETE). Call ADDR to calculate the disk word address of the designated job in LJD.

- Call DWRITE to write a word of zeros in 1st word of LJD entry to indicate an empty entry.

- Decrement job count by one, and go to step c.

g. (MODIFY). Call ADDR to calculate the disk word address of the designated job in LJD.

- Adjust disk word address to address of 4th and 5th words of LJD entry.

- Call DWRITE to rewrite 4th and 5th words of LJD entry with input data.

h. (EXIT). Save Q. If function code = 0 (enter new job) or 1 (locate job by Job Name), release input buffer.

- Call EXDFUN to release GJB.

- Exit to EXTSR to return to caller of GETLJD, and with Q and A set.

INTERFACE

1. ENTRY METHOD

    GJB is a disk resident program.

    Entry is via ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

    (I) = Base address of Active Terminal Table (ATT)

    (ATT/ATEMP) = Address of parameter list

word			
0	}	Six character }	Function Code = 0 or 1
1	}	Job Name }	
2	}		

0	Verification Options/Maximum Record Length -1 }	Function Code
1	Magnetic Tape Format Number/EOB Own Code Number }	= 4

3	Verification Options/Maximum Record Length -1 }	Function Code
4	Magnetic Tape Format Number/EOB Own Code Number }	= 0
5	Function Code	
6	Job Index Function Code = 2, 3, or 4	
7	Job Count Value set by GJB	

3. EXIT METHOD

    Exit is via EXTSR to the caller of GETLJD.

4. OUTPUT PARAMETERS OR CONDITIONS

    (I) = Base address of ATT

    (Q) = Return code

        = 0   Designated job name does not exist in LJD.

        = 1   Requested function was executed.

        = $FFFF LJD is full; new job cannot be entered.

    (A) = Job index corresponding to designated job.

    Word 7 of Parameter List = Current Job Count

    If Function Code = 1, words 3 and 4 of Parameter List are filled from LJD entry.

    If Function Code = 2, words 0-4 of Parameter List are filled from LJD entry.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK ($2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        FIVE ($43)
        SST ($47)
        CLCDWA ($90)
        ENTSR ($91)
        EXTSR ($92)
        EXDFUN ($9F)
        REDSEC ($A7)
        REDDWA ($A9)
        WRTDWA ($AA)
        CAM ($AE)
        WAIT ($B0)

    b. SST

        EXTSST (0)
        CATT (11)

    c. EST

        LJDDA (8)

51

d. ATT

ATEMP (73)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

ENEXSR
{
ENTSR   Save return address in Exits stack.
EXTSR   Exit to last return address in Exits stack.
SETMOV   Enable moving of disk resident program.
WAIT   Schedule relay address.

ENTPRG   (EXDFUN) Exit disk function.

PIO
{
CLCDWA   Calculate disk word address.
REDSEC   Read a sector.
REDDWA   Read disk word address.
WRTDWA   Write disk word address.

CAM   Core allocation module.

4. DATA AREAS USED

SST   System Status Table
EST   Extended System Table
ATT   Active Terminal Table

Local Data Area

JOBCNT   Job Counter. Set to total number of jobs in LJD.
Used in searching LJD.
EMPTY   Job Index of first empty LJD entry.
JCOUNT   Address of word 7 in parameter list.
PTSAVE   Address of LJD entry located by GETJOB.
JDX   Current Job Index value.
SEXIT   Address of word 0 of parameter list.
NAMPTR   Address of word 2 of parameter list.

5. LOCAL SUBROUTINES

a. GETJOB

This subroutine searches the LJD for an entry with the same alphanumeric job name as the entered job name.

On entry, SEXST = Address of Job Name

On exit, JDX   = LJD Index if job found

Q = 0 if not found
Q = 1 if found
Q = 0 if not found and LJD full.

DOCUMENT CLASS __GJB_____

PROGRAM NAME____Maintain Legal Job Directory_____

The routine requests a buffer of one sector and reads into it the LJD one sector at a time. Then each section is searched for a match. The loop is controlled by the LJD job count, with extra tests for no jobs and table full. It has its own subroutines - SECTOR - to read down the required areas.

b. DWRITE

This subroutine writes to disk according to parameters set up by the caller.

c. ADDR

This subroutine converts a job index to a disk word address using CLCDWA.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$EC_{16} = 236_{10}$ words

2. RELOCATION TYPE

Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

DOCUMENT CLASS ___GSE___

PROGRAM NAME___Get/Store character; General error exit___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   GSE has three entry points:

   GBYTE extracts the current character from the Terminal Record Buffer (TRB).

   SBYTE stores the current character in the current position in the TRB.

   ERROR displays the designated error code and performs cleanup operations on Terminal Input Queue (TIQ) and Active Terminal Table (ATT).

2. DETAILED DESCRIPTION

   a. GBYTE

      ● Calculate word address of current character in TRB:

        $$\text{Address} = (\text{ATT/TRB}) + 8 + \frac{(\text{ATT/RCNT})}{2}$$

      ● Isolate current character in bits 0-6 of A-register, with bits 7-15 of A=0.

      ● Exit to caller.

   b. SBYTE

      ● Calculate word address of current character in TRB:

        $$\text{Address} = (\text{ATT/TRB}) + 8 + \frac{(\text{ATT/RCNT})}{2}$$

      ● Move the current character from ATT/CURCH/0-7 to the current position in TRB.

      ● Exit to caller.

   c. ERROR

      1. (ERROR). Save error index in ATT/ECODE.

         ● If mismatch error, clear ATT/MISCH.

         ● If ERROR was entered from a disk resident program, call EXDFUN to release the program, if any.

         ● If ATT/ECODE 8 (DATA type error, Sign missing, Illegal Character, Illegal function key, character mismatch, Beginning of data, End of data), or ATT/ECODE = $15 (own code not on disk), go to step 2.

         ● If ATT/WQI/0 = 1 (cancel or Err O/R not required), go to step 2, otherwise go to step 3.

         ● (NOTVAL). If ATT/SW2/b=1, perform REGEN, and clear switch. Else, display error code.

         ● Set ATT/SW1/7 = 1 (Reset or Disp. key required).

         ● If ATT/RCNT ≠ 0 (not at beginning of record), set ATT/SW1/0 = 0 (Cancel or Err/R key not required).

DOCUMENT CLASS ___GSE_____

PROGRAM NAME_____Get/Store character; General error exit_____

- Calculate relative address of TIQ: Rel. Addr. = (Terminal No.) x (Input Buffer Length).

- Zero out first word of TIQ for current terminal (Set character count = address of next char. out = 0), with interrupt system disabled.

- Set ATT/0/12/13,14 = 0 (System lock, Pseudo function lock, Function lock), and set ATT/0/11 = 1 (Error lock).

- Set ATT/SW2/8 = 0 (LL error is not to be displayed).

- Call DISPLY to display the error code represented in ATT/ECODE.

- Move current character from ATT/CURCH/0-7 to ATT/ECODE/8-15.

- Set ATT/TRMNT/15 = 0 (Current function cannot be terminated by Cancel key).

- Exit to CLRSR.

2. (AUTOFG). If Automatic error flagging is not selected for the current field, go to step 2.

- If it is selected, set ATT/ECODE = 0, and call ENTPRG to load and execute ERRKEY, entry point 1 (OK), with no return.

INTERFACE

1. ENTRY METHOD

    a. GBYTE

```
EQU   GBYTE ($94)
RTJ-  GBYTE
--Return--
```

    b. SBYTE

```
EQU   SBYTE ($95)
RTJ-  SBYTE
--Return--
```

    c. ERROR

```
EQU   ERROR ($96)
ENQ   Error Index (See section, PHYSICAL CHARACTERISTICS)
JMP-  ERROR
```

    In each case, (I) = Base address of ATT

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

130

DOCUMENT CLASS ___GSE___

PROGRAM NAME___Get/Store character; General error exit___

        LPMASK (2)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        CLRSR ($93)
        ENTPRG ($9E)
        EXDFUN ($9F)
        DISPLY ($A0)

    b. SST

        FFA (12)
        IBL (14)
        TIQ (15)
        STEMP (20)

    c. ATT

        CURCH (2)
        RCNT (4)
        ATRB (6)
        SW1 (19)
        SW2 (20)
        FUNIDX (21)
        TRMNT (23)
        ECODE (34)
        BMODE (15)
        MISCH (5)

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    ENEXSR   (CLRSR)   Clear Exits stack and Function Lock
    ENTPRG   (ENTPRG)   Load/Execute a disk resident program
              (EXDFUN)   Exit disk resident program
    DISPLY   Display error code

4. DATA AREAS USED

    SST   System Status Table
    ATT   Active Terminal Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $67_{16} = 103_{10}$ words

DOCUMENT CLASS ___GSE_____

PROGRAM NAME___Get/Store character; General error exit_____

2. RELOCATION TYPE

   Core-resident, loader relocatable

3. ENTRANCY CLASSIFICATION

   GBYTE and SBYTE are Reentrant
   ERROR is dormant reentrant

ERROR CODES

Index$_{16}$	Code	Meaning
0	--	Clear error code
1	A1	Data Type error
2	A2	Sign missing
3	A3	Illegal character
4	A4	Illegal function key
5	A5	Mismatch character
6	B1	Beginning of data
7	B2	End of data
8	C1	Limit error
9	C2	Special test error
A	C3	Check digit error
B	C4	Own code test error
C	C5	Nonzero error
D	D1	Invalid operator no.
E	D2	Invalid format no.
F	D3	Invalid document no.
10	D4	Invalid job name
11	D5	Invalid batch no.
12	E1	Illegal request
13	E2	Illegal parameter
14	F1	Format no. in document error
15	F2	Own Code is not on disk
16	F3	Disk full
17	F4	Core full try again
18	F5	Function busy try again
19	LL	System lock - wait
1A	PP	Parity error
1B	**	Field with error flag
1C	C6	Balancing error
1D	D6	Invalid auto sequence count
1E	F6	Disk down
1F	F7	Batch full
20	SM	Message from Supervisor

/3/

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___HDRALP - Display Header Information___ Level 06___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   HDRALP has 5 entry points:

   CMOD displays a single character (V, R, or E) to indicate the current mode.

   CJOB displays the current job name.

   CERR displays an error code designated by an input parameter.

   CMLDS displays M or space, depending on whether multilevel Dup/Skip key was entered.

   CSUP displays -SV- on the screen.

2. DETAILED DESCRIPTION

   a. CMOD

      1) (CMOD). Call ENDSR to save return address in Exits stack.

         ● Using ATT/MODE/8-10 (mode) as an index, move a character from MODTABL into DATA area of CDATA request to indicate the current mode of operation:

                V: Verify
                R: Read
                E: Entry

      2) (COMEX). Call CONCSR to convert screen address for mode field to codes for CDATA REQUEST.

      3) (JOBEX). Call CDATA to display DATA area.

         ● Exit via EXDSR

   b. CJOB

      ● Call ENDSR to save return address in Exits stack.

      ● Move 6-character Job name from Terminal Output Table (TOPT) to DATA area of CDATA request.

      ● Set the fourth word of the parameters list to FFFF to indicate end-of-data.

      ● Go to step a3 to display job name and exit.

   c. CERR

      ● Call ENDSR to save return address in Exits stack.

      ● If TOPT/BLOKS/4-6 = 7 (System lock waiting to be displayed), set TOPT/BLOKS/4-6 = 0.

o Move 2-character error code, followed by Bell code, to DATA area of CDATA request.

o If ATT/RQ + 1 = 0 (Clear error code), call CONCSR to convert current screen position to codes for CDATA request.

o If ATT/MODE/8, 12, 13 ≠ 0 (Entry, Verify, Pseudo-Entry, or Pseudo-Verify mode), advance screen position by 1 character.

o Go to step a3 to clear/display error code.

d. CMLDS

o Call ENDSR to save return address in Exits stack.

o If ATT/CURCH/0-7 = $77 (Multi-Level Dup/Skip key) and ATT/RCNI ≠ 0 (beginning of record), call TUTMOD to write tutorial display. The display is written only if tutorial mode has been requested (AID command).

o If ATT/SW1/6 = 1 (All auto fields to be entered manually until next Multilevel Dup/Skip key), insert M into DATA area of CDATA request, to display in header line.

o If ATT/SW1/6 = 0, insert space.

o Call CONCSR to convert current screen position to codes for CDATA request.

o Go to step a3 to display M or space.

e. CSUP

o Call ENDSR to save return address in Exits stack.

o Insert -SV- in DATA area of CDATA request.

o Call CONCSR to convert interrogate line position to codes for CDATA request.

o Go to step a3 to display -SV-.

INTERFACE

1. ENTRY METHOD

   EXT   P   where P = CMOD, CJOB, CERR, CMLDS, or CSUP.

   RTJ   P

   -Return-

2. INPUT PARAMETERS OR CONDITIONS

   (I) - Base address of Active Terminal Table (ATT)

   When CERR is called, (ATT/RQ + 1) = Error Code Index

3. EXIT METHOD

    Return to location immediately following RTJ P.

4. OUTPUT PARAMETERS OR CONDITIONS

    (I) = Base address of ATT

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK   (2)

        ONEBIT   ($23)

        CDATA   ($8D)

        ENDSR   ($8E)

        EXDSR   ($8F)

    b. ATT

        MODE   (2)

        CURCH   (2)

        RCNT   (4)

        SW1   (19)

        TOPT   (33)

        RQ   (65)

    c. TOPT

        CURSRD   (3)

        JOB   (5)

        BLOKS   (14)

    d. KEY CODE

        MLDSCH   ($77)

2. MACROS USED

    None

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME_____HDRALP - Display Header Information_____Level 06_____

3. PROGRAM DEPENDENCIES

VL2CRT   (CDATA)   Display a message

CMISCI $\begin{cases} \text{ENDSR} \\ \text{EXDSR} \end{cases}$ Enter display subroutine
                     Exit display subroutine

4. DATA AREAS USED

ATT   Active Terminal Table

TOPT   Terminal Output Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$77_{16} = 119_{10}$ words.

2. RELOCATION TYPE

Core resident, loader relocatable.

3. ENTRANCY CLASSIFICATION

Passive reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___HDRNOS___   Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    HDRNOS contains routines which update various fields on the screen as follows:

        CFMT displays the current format number.

        CRCNT displays the record count.

        CCOL displays the current column number.

        CFNDTY displays field number and data type.

        CDOC displays document number.

        CBAX displays batch number.

        CFDNO displays field number at beginning of line.

        CHDR regenerates the entire CRT header display (all the above, except
          CFDNO, and job name, mode, error code and multi-level SKIP/DUP).

    The following entry points are also defined and these represent the screen
    addresses of various fields.

        SVADR   0   supervisor

        JOBADR   0   job name

        BAXADR   7   batch address

        MODADR   11   mode

        DOCADR   13   document number

        FMTADR   17   format number

        RECADR   22   record number

        FLDADR   27   field number

        DTYADR   29   data type

        COLADR   33   column number

        ERRADR   38   error code

        MDPADR   46   multi-level SKIP/DUP

2. DETAILED DESCRIPTION

    a. CMFT entry point to display format number.

        1) Enter display routine via ENDSR, set request cell to screen address
            for format number and get format number from ATT. If format number
            is 0, go to e3 (display blanks).

2) Set digit count to 4.

3) Save display stack pointer. Call BD to convert value to BCD. If there are 4 digits, go to a5.

4) Shift BCD value left 1 digit.

5) Save display address of column number. If not displaying column number, go to a8. Save previous value of column number in TOPT and replace with new column number.

6) Compare digit of new column number with previous value. If they are not equal, go to a8. If equal, increment loop counter. If all digits have been checked, clear control flags and exit via EXDSR.

7) If there are more digits, increment the request cell (screen address), shift new and old values left 1 digit and go to a6.

8) Set indicator for upper byte and save data block address.

9) Pick up word of BCD characters, shift next digit to low order nybl and save. Mask off all but low nybl. If indicator is set for lower byte, go to a12.

10) Move BCD character to upper byte, convert to ASCII and add fill character in low byte. Store value in display area.

11) Invert the byte indicator and increment the loop count. If there are move digits, go to a9. If done, go to a13.

12) Clear fill character out of lower byte of current word in display and insert new character, converting it to ASCII. Increment display pointer and go to a11.

13) If data type flag is not on, go to a14. If on, get field type from ATT and pick up corresponding display word from data type table. Go to a15.

14) If slash flag is set (field number) go to a16. Otherwise go to a17.

15) Convert low byte of current display word from filler to blank and increment pointer. Store display word (from table) into TOPT. Go to a18.

16) Get current display word and convert filler to slash.

17) If indicator is currently set for high order byte, go to a19.

18) Increment display pointer.

19) Mark end of display with FFFF.

20) Clear all flags. Get screen address and call CONCSR to convert it to O/P format and call CDATA to initiate display driver. Exit via EXDSR.

b. CRCNT entry point to display record count.

1) Enter display routine via ENDSR, set request cell to screen address
    for record count. Pick up current mode from ATT. Check mode - if
    not Read mode go to b3.

2) Get number of records in read mode from ATT and go to b4.

3) Get total number of records from ATT.

4) Increment count by 1. Go to a2 (common display code).

c. CCOL entry point to display column number.

1) Enter display routine via ENDSR, set column number flag, set request
    cell to screen address for column number. Get character count for
    current record from ATT. Go to b4.

d. CFNDTY entry point to display field number and data type.

1) Enter display routine via ENDSR and set request cell to screen address
    for field number. Set flag for data type. Get field number from ATT.

2) Increment current value by 1.

3) Set counter for 3 digits and go to a3 (common display code).

e. CDOC entry point to display document number.

1) Enter display routine via ENDSR and set request cell to screen address
    for document number. Get SW1 from ATT and isolate indicator for docu-
    ment or format select. If indicator is set (no legal document number),
    go to e3 (display blanks).

2) Get current mode from ATT. If mode is entry, go to e5.

3) Set data block to 2 words of blanks. Set 3^d word of data block to
    FFFF (end of block indicator). Go to a20 (common display and exit
    point).

4) Get document number from ATT and go to d3 (3 digit field code).

f. CBAX entry point to display batch number.

1) Enter display routine via ENDSR and set request cell to screen address
    for batch number. Get BCD batch number from ATT, set digit count for
    3 and go to a4.

AA5641

g. CFDNO entry point to display field number at beginning of line.

1) Enter display routine via DNSR. Get TOPT adds from ATT and get
    current cursor adds from TOPT. Add 1 and set in request cell
    (i.e. next screen address). Set current cursor address in TOPT
    to 3. Set screen address of current data field in TOPT to cursor
    address value + 3. Set slash flag. Get field number (passed in
    ATT) and go to d2 (common code to increment value and set digit
    count to 3).

i. CHDR entry point to regenerate CRT header display.

1) Enter display routine via ENDSR. Call CJOB to display job name,
    CBAX (6) to display batch number, CMOD to display mode, CDOC (5)
    to display document number, CFMT (1) to display format number,
    CRCNT (2) to display record count, CFNDTY (4) to display field
    number and data type and CCOL (3) to display column number.

2) Get ERRTAB index from ATT for error code and store in the ATT.
    Call CERR to display the error code and CMLDS to display multi-
    level SKIP/DUP. Exit via EXDSR.

INTERFACE

1. ENTRY METHOD

    RTJ

2. INPUT PARAMETERS OR CONDITIONS

    For all entry points, I = address of current ATT.

    For CFDNO, field number must be passed in RQ of the ATT.

3. EXIT METHOD

    All routines exit to the caller via JMP (EXDSR).

4. OUTPUT PARAMETERS OR CONDITIONS

    None

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___HDRNOS___   Level 06

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK   (2)

        NZERO   ($12)

        ONEBIT   ($23)

        SST   ($47)

        ENDSR   ($8E)

        EXDSR   ($8F)

        CDATA   ($8D)

    b. SST

        TEMP   (20)

        SA   (SST + TEMP + 1)

        SQ   (SST + TEMP + 2)

        TMP   (SST + TEMP + 2)

        COUNT   (SST + TEMP + 3)

    c. ATT

        MODE   (2)

        COL   (2)

        CURSRD   (3)

        RCNT   (4)

        FDTP   (13)

        FMTNO   (13)

        FLDN   (14)

        FLDST   (15)

        DOCNO   (17)

        SW1   (19)

        RCDRD   (31)

        RDC   (32)

AA5641   PRINTED IN THE USA

TOPT   (33)
ERCODE   (34)
BATCH   (35)
RQ   (65)

d. RQ
    C   (1)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

CMOD   subroutine to display mode
CJOB   subroutine to display job name
CERR   subroutine to display error code
CMLDS   subroutine to display multilevel SKIP/DUP
CONCSR   subroutine to convert screen address from binary to O/P format
BD   subroutine to convert binary to BCD
ENDSR   enter display subroutine (via transfer vector)
EXDSR   exit display subroutine (via transfer vector)
CDATA   CRT driver initiator (via transfer vector)

4. DATA AREAS USED

LOCORE
ATT   (current ATT address is input in 1 and data to be displayed is
          retrieved from the ATT)
TOPT   (TOPT located via ATT)
SST   temp area

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

EA_{16} words.

2. RELOCATION TYPE

   Loader relocatable.

3. ENTRANCY CLASSIFICATION

   Dormant reentrant.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ IDKCH - IDK Character/Function Processor   Level 6___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    IDKCH is called by a direct jump from MPC when it cannot anticipate
    what input is next, to determine whether the current character is
    valid or not.

    IDKCH exits to the appropriate processing routine, or to ERROR, if
    an error condition exists.

2. DETAILED DESCRIPTION

    a. (IDKCH). If (Q) = 0 (current key is a function key), go to step g.

        ● If any of ATT/SW1/0,3,5,7,8,9, or 11 = 1 (a specific function
          key is required), go to step c.

        ● If ATT/SW2/12 = 1 (ERR 0/R key must be next), go to step c.

        ● Call CHIDLE to check for Idle mode (step t). CHIDLE does not
          return here if in Idle mode.

        ○ (Mode is Verify). If ATT/MISCH = 0 (no mismatch character),
          go to step d.

        ● Set ATT/MISCH = 0.

    b. (IDK42). Set ATT/CMODE/14 = 0 (IDK mode) and go to step f.

    c. (ERR). Exit to ERROR with (Q) = CA3 (illegal character).

    d. (IDKCH4). Nonfunction character has been entered in Entry mode
       or in Verify mode with no mismatch condition.

        ○ If ATT/SW1/2 = 1 (Positioned at beginning of record) go to
          step e.

        ○ If ATT/SWP/11 = 1 (no forward motion allowed) exit to ERROR
          with (Q) = CB2 (end of DATA).

        ● Otherwise go to step c.

    e. (IDKCH1). Call BORENT to perform beginning of record housekeeping.

    f. (IDKCH3). Nonfunction character has been entered in Entry or
       Verify mode.

        ● Exit to CONVRT in order to convert the entered character according
          to the data type and to continue normal processing of an
          alphanumeric character.

g. (IDKFUN). Function key has been entered.

- If ATT/SW1/7 = 0 (RESET or DISP key are not required) go to step h.

- If RESET was entered, go to step u.

- If DISP was entered, and ATT/0/11 = 1 (Error Lock) go to step u.

- Otherwise go to step j.

h. (CHMISM). Call CHIDLE (Step t) to check for Idle mode. (CHIDLE will not return if in Idle mode.

- If ATT/BMODE/13 = 1 (Entry mode), go to step k.

- If ATT/MISCH = 0 (no mismatch character in Verify mode) go to step k.

- If current character is $20 (space), SKIP key, DUP key, or READ key, go to step u.

i. (RPL). If current character is RPLFLD key or COR key, go to step u.

j. (ERRFUN). Exit to ERROR with (Q) = CA4 (Illegal Function key).

k. (CHCOR). Function code has been entered where reset is not required and the station is in Entry mode or in Verify mode with no mismatch condition.

- If ATT/SW1/9 = 0 (SKIP key not required) go to step m.

- If current character is SKIP key, go to step 1.

- If current character is RPLFLD key, go to step u.

- If current character is not REL key, go to step j.

- If current character is REL key, and ATT/CMODE/13 = 1 (Pseudo Entry mode) go to step j.

- If ATT/SW2/13 = 1 (Revalidate), go to step j.

- Otherwise set (Q) = $FFFF (current character is REL key).

l. (CLEAR). (Current character is SKIP key or REL key).

- Set ATT/SW1/9 = 0 (SKIP key not required).

- Set ATT/CMODE/14 = 0 (not IDK mode).

- If (Q) = 0 (REL key) exit to VALID.

- If ATT/CMODE/12 = 1 (Pseudo Verify mode), or ATT/CMODE/9 = 0 (Verify mode) exit to ENDOF1 to perform the processing of End of Field in Verify mode.

- In Entry mode or Pseudo Entry mode exit to EMM1 to perform Validation and any other processing of End of Field in Entry mode.

m. (ONCOR). Function code has been entered where Reset is not required and the station is in Entry mode or in Verify mode with no mismatch condition, and SKIP key is not required.

- If ATT/SW1/11 = 1 (COR key must be next), go to step v.

- If ATT/SW1/5 = 0 (CONT of Sight Verification not required), go to step n.

- If current character is CONT key, RPL FLD key, BSC key, REL key, space, or DLT key, go to step u.

- Otherwise go to step s.

n. (CHCDOC). If ATT/SW1/3 = 0 (format assignment is not required) go to step o.

- If current character is DOC key, FORM key, FORM SEL key, or RTRN key, go to step u.

- Otherwise go to step j.

o. (CHEOB). If ATT/SW1/0 = 0 (no EOB validation error) go to step p.

- If current character is CANCEL key, go to step u.

- Otherwise, go to step q.

p. (CHEOVR). If ATT/SW1/12 = 0 (ERR O/R key not required) go to step r.

- If current character is RPLFLD key, go to step u.

q. (IDF8). If current character is ERR O/R key, go to step u.

- Otherwise go to step j.

r. (CHFORM). If ATT/SW2/11 = 0 (forward motion allowed) go to step u.

- If ATT/SW1/12 = 1 (record insertion), go to step u.

s. (IDF9A). If current character is INT key or READ key, go to step u.

- Otherwise go to step j.

t. (CHIDLE). If not in Idle mode, return to caller.

- If current character not a function key, or if it is a space, call DISP to display current character, and exit to CLRSR.

- If current character is INT key go to step u.

- Otherwise, set ATT/CURCH/0-7 = $40 (Set current character to @, call DISP to display current character, and exit to CLRSR.

u.   (FUNOK). Exit to ENTFUN.

v.   (CORNXT). If ATT/SW1/12 = 0 (no record insertion) go to step w.

   ● If current character is DLT key, go to step u.

   ● Otherwise, go to step x.

w.   (CHCDEL). If ATT/SW1/1 = 0 (not COR/DLT key), go to step i.

   ● If current character is CANCEL key, go to step u.

x.   (PURCUR). If current character is COR key, go to step u.

   ● Otherwise, go to step j.

INTERFACE

1. ENTRY METHOD

    IDKCH is entered by a jump from MPC (Main Process Control)

2. INPUT PARAMETERS OR CONDITIONS

    (Q) = 0 if current character represents a function key (including Space).
        ≠ 0, Q contains nonfunction character code.
    (I) = Base address of Active Terminal Table (ATT)

3. EXIT METHOD

    IDKCH may exit to any of several routines, depending on conditions:

    CONVRT if a nonfunction character was entered at beginning of a
    record.

    EMM1 if SKIP key (for boundary control) was entered in Entry mode or
    Pseudo Entry mode.

    ENDOF1 if SKIP key was required and had been entered in Verify or
    Pseudo Verify mode.

    VALID if REL function was entered when SKIP was required.

    CLRSR is a key other than INT key was entered in Idle mode.

    ERROR if illegal character, end of data, or illegal function.

    ENTFUN if valid function.

4. OUTPUT PARAMETERS OR CONDITIONS

    (I) = Base address of ATT

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____IDKCH - IDK Character/Function Processor_____Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

       LPMASK (2)
       NZERO ($12)
       ZERO ($22)
       ONEBIT ($23)
       ZROBIT ($33)
       CLRSR ($93)
       ERROR ($96)
       DISP ($A0)

    b. ATT

       CMODE (2)
       CURCH (2)
       FCNT (3)
       MISCH (5)
       BMOD (15)
       SW1 (19)
       SW2 (20)
       ATTTMP (73)

    c. KEY CODES

       BSC ($1A)
       DUP ($1C)
       SPC ($20)
       AT ($40)
       RES (67)
       SKIP ($68)
       FB ($6D)
       FA ($6E)
       REL ($6F)
       DOC ($71)
       DSPKEY ($ 73)
       RPLFLD ($74)
       CONT ($75)
       CANCEL ($76)
       DLT ($79)
       ERROVR ($7A)
       FS ($7B)
       COR ($7C)
       RTD ($7D)
       READ ($7E)
       INT ($7F)

d. ERROR CODES

CA3 ($3)
CA4 ($4)
CB2 ($7)

e. EXTERNALS

BORENT
CONVRT   Entry point in MPC
AUTO
ENTFUN   Entry point in MPC
DCHAR   Parameter for DISPLAY
EMM1   Entry point in EMM
ENDOF1   Entry point in VMM
VALID   Entry point in SKPDUP

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

BORENT   BORENT   Beginning of Record procedure
MPL   CONVRT   Process an input character
           ENTFUN   Process a function key
EMM   EMM1   Validate current field and perform other end of
                          field Entry mode
VMM   ENDOF1   End of field - Verify mode
SKPDUP   VALID   REL key processing

4. DATA AREAS USED

ATT   Active Terminal Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$D5_{16} = 213_{10}$ words

2. RELOCATION TYPE

IDKCH is core-resident, loader-relocatable.

3. ENTRANCY CLASSIFICATION

Reentrant code.

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME___INSERT_____Level 06_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    INSERT = sets up parameters when INS is keyed on a station.

2. DETAILED DESCRIPTION

    a. If not pure read on 'ILLEGAL FUNCTION KEY' error message is generated.
        The same error occurs if INSERT was not keyed at the beginning of the
        record.

    b. If the basic mode is Entry, check that total number of records in entry
        is equal to number of records in Read. If they are equal 'illegal func-
        tion key' error message is generated.

    c. Check according to SW1, BIT 13 if the last record needs to be rewritten.
        If so, rewrite the last record.

    d. Set insert and BOR switches, set IDK and PSEUDO ENTRY bits.

    e. Reset the screen to get a new record and exit through CLRSR.

INTERFACE

1. ENTRY METHOD

    Entered by MPC upon detection of that key function, via ENTPRG.

2. PARAMETERS

    I = ATT base address

3. EXIT METHOD

    Either to ERROR or via CLRSR

4. OUTPUT PARAMETERS

    None

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK   (2)

        ZERO   ($22)

DOCUMENT CLASS **CYBERDATA**

PROGRAM NAME **INSERT**   Level 06

           ZROBIT   ($33)

           NZERO   ($12)

           ONEBIT   ($23)

   b.   ATT

           CMODE   (2)

           RCNT   (4)

           ATRB   (6)

           BRCNT   (15)

           BMODE   (15)

           SW1   (19)

           PCDRD   (31)

           RCD   (32)

   c.   TRANSFER VECTOR

           ERROR   ($96)

           CLRSR   ($93)

           REWRIT   ($A4)

           DISPLY   ($A0)

   d.   EXTERNALS

           DEOREV

   e.   DATA AREAS USED

           ATT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $27_{16} = 39_{10}$ program length.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____INTRGT_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   INTERROGATE key function program. It receives the character string which
   follows the INTERROGATE key depression. INTRGT passes control to
   INTERROGATE functions.

2. DETAILED DESCRIPTION

   INTRGT starts with a call to SETMOV for clearing the I/O bit. If the
   station is in Idle mode, DISPLAY is called in order to clear the screen.
   Data type is set to ALPHA, and GETSTR is invoked for reading a character
   string. On return, if a format exists, DT will be set back to old data
   type. Then checks are made in the INTERROGATE buffer if the entered
   key was RELEASE or CANCEL, or if there was an overflow. In the last case
   ERROR will be called with error code $13.

   If CANCEL was entered, INTRGT exits through EXDFUN and CLRSR.

   If RELEASE was detected, the function index is calculated from the name
   by the procedure FINDEX. If the function index is illegal, ERROR is
   called with error code $12, and with bit 8 Switch 1 set. INTRGT checks
   the mode and if it is Idle mode, checks for the functions ENT, VER, THW
   or SUP; other functions are illegal in that mode, and error is called with
   error code $18. ENTPRG is invoked for loading the disk program for
   performing the function. If there is not enough core for loading the
   function, INTRGT will exit to WAIT and try again, according to the load
   counter, exiting to ERROR with error code $23 if core was not available
   after n cycles.

   The internal procedure FINDEX:

   Calculates function index according to IFTAB, where all function names
   are stored in the form $A*P1+B*P2+C$, where: ABC - is function name
                                          $P1 = \$400, P2 = \$0020.$

INTERFACE

1. ENTRY METHOD

   INTRGT is loaded via ENTPRG, when the operator hits at the only entry
   point INTRGT.

   Execution begins at the only entry point INTRGT.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

3. EXIT METHOD

    JMP - (ERROR)   - with error code in Q, when error occurs.
    RTJ - (EXDFUN)
    JMP - (CLRSR)   - when CANCEL is entered.
    RTJ - (EXDFUN)
    JMP + to entry point in an INT. FUNCTION.
        program - when RELEASE key is entered and the comand is legal.

4. OUTPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        CLRSR ($93)
        WAIT ($B0)
        DISPLAY ($A0)
        ERROR ($96)
        ENTPRG ($9E)
        EXDFUN ($9F)
        GETSTR ($A2)
        SST ($47)

    b. SST

        STEMP (20)
        TSTAB ($8A)

    c. ATT

        FBA (7)
        DT (13)
        RFA (14)
        MODE (2)
        SW1 (19)
        PI (21)
        IRB (27)
        ATEMP (73)

    d. CONSTANTS

        ERR1 ($13)
        ERR2 ($4)
        ERR3 ($12)
        ERR4 ($12)
        ERR5 ($17)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

SETMOV	–	Clears I/O bit to allow program move.
DISPLAY	–	SETDIS – Set INT. display bit.
		CLRALL – Clears screen.
		CLRFLD – Clears a line.
		DMOD – Display mode.
		REGEN – Regenerates screen.
CLRSR	–	Clears address address stack and exits to DISPATCHER.
WAIT	–	Schedules the program and exits to DISPATCHER.
ERROR	–	Displays error codes and exits.
ENTPRG	–	Loads a disk resident program into core.
EXDFUN	–	Decrements user count for program.
GETSTR	–	Reads a character string into interrogate buffer.

4. INTERNAL SUBROUTINES

    FINDEX –   Finds a command name in the table (IFTAB) and calculates function index.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $(FC)_{16} = (252)_{10}$.

2. RELOCATION TYPE

    Run anywhere, movable.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME____IXPL_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   IXPL is interrogate function to display the layout of any interrogate function.

   Calling sequence is -
   >   XPL,aaa where aaa is interrogate function mnemonic

   The function layout texts are contained within the program

2. DETAILED DESCRIPTION

   IXPL first enters SETMOV to enable program moving and then checks if the
   requesting station is 970-480. When the station is not 970-480 goes to
   set regen bit (ATT/SW1/8), and takes ERROR exit to indicate illegal request
   (CMDERR). If it is 970-480, checks the character following the XPL request,
   if it is not a comma goes to CMDERR. If it is a comma, collects the 3
   ASCII characters of the requested function mnemonic through GETCHR
   subroutine, packs them into one word key and looks for end of record.
   If end of record is not found goes to CMDERR. Else performs a loop to
   check if there is a corresponding entry in the keys table to the requested
   function key and when there is no entry to CMDERR. If there is a
   corresponding entry requests an interrogate buffer for an output line and
   moves the corresponding message from the texts area to the output line.
   When finished, issues a display request, releases itself via EXDFUN,
   and exits to dispatcher via CLRSR.

INTERFACE

1. ENTRY METHOD

   IXPL is an interrogate function loaded by ENTPRG.
   Only entry point in XPL.

2. INPUT PARAMETERS OR CONDITIONS

   I ▬ current ATT address

   Operator enters -
   >   XPL,aaa

3. EXIT METHODS

   a. Jump to ERROR with appropriate error code.

   b. Jump to CLRSR to clear the stack and exit to dispatcher.

4. OUTPUT PARAMETERS OR CONDITIONS

   I = Current ATT address.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __IXPL__   __Level 6__

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)

    b. SST

        STEMP (20)

    c. TRANSFER VECTOR

        CLRSR ($93)
        ERROR ($96)
        GETALF ($9B)
        GETCOM ($9C)
        EXTDFUN ($9F)
        DISPLY ($A0)

    d. ATT

        SW1 (19)
        AINT (27)
        RQ (65)
        ATEMP (73)

    e. CONSTANTS

A	($1)	P	($10)
B	($2)	Q	($11)
C	($3)	R	($12)
D	($4)	S	($13)
E	($5)	T	($14)
F	($6)	U	($15)
G	($7)	V	($16)
H	($8)	W	($17)
II	($9)	X	($18)
J	($A)	Y	($19)
K	($B)	Z	($1A)
L	($C)	AA	($41)
M	($D)	AZ	($5A)
N	($E)	C1	($40)
O	($F)	C2	($20)

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __IXPL__   Level 6

2. PROGRAM DEPENDENCIES

    ENEXSR - (SETMOV) - enable program moving.
             (CLRSR) - clear stack and exit to dispatcher.
    ERROR   -   - error condition processor.
    SCANV   - (GETALP) - get alpha character from input string.
             (GETCOM) - get next comma or end of record.
    ENTPRG - (EXDFUN) - release disk resident program.
    PIO   - (GETBUF) - get interrogate buffer for a terminal.
    DISPLY -   - handle display output.
             (DLINE) - display line on 970-480
             (CLRFLD) - clear field on screen.

3. DATA AREA USED

    ATT
    SST

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $111_{16} = 273_{10}$ words.

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____KDMY - Key Dummy Function_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   KDMY is the dump program for key functions.

2. DETAILED DESCRIPTION

   KDMY jumps to ERROR (in GSE) with Q register containing error code $1B
   (program not on disk).

INTERFACE

1. ENTRY METHOD

   KDMY is loaded by ENTPRG. When ENTPRG does not find on disk the program
   to process the required key function, a call is made to load KDMY, the
   dummy key function program.

2. INPUT PARAMETERS

   None.

3. EXIT METHOD

   KDMY jumps to ERROR (GSE) with no return.

4. OUTPUT PARAMETERS OR CONDITIONS

   Error message is displayed.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   SST:

   ERROR ($96)

2. PROGRAM DEPENDENCIES

   GSE (ERROR) Display error code.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE

   7_{10} words.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___LBLO - Routine to Write Unlabelled Tapes_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   LBLO has 4 entry points. VOLHDR rewinds the tape. FILHDR does nothing.
   FILTRL writes a tape mark. VOLTRL writes a tape mark and rewinds. Thus,
   the only effect is to separate files of data blocks by a tape mark.

2. DETAILED DESCRIPTION

   a. VOLHDR Entry: After calling SETMOV to allow the program to move,
      TAPIO is called to rewind the tape. The program is released with a
      call to EXDFUN; and the next overlay is called by loading Q with the
      function index in ATT/MTFEXT and calling ENTPRG.

   b. FILHDR Entry: SETMOV is called, followed immediately by calls to
      EXDFUN and ENTPRG.

   c. FILTRL Entry: The SETMOV call is followed by a call to TAPIO to write
      a tape mark.

      If this is not end-of-tape or ATT/MTFENT = $7FFF (entry from TRC),
      exit with calls to EXDFUN and ENTPRG. Otherwise call PRTMSG to print
      "END OF TAPE", and jump to EOV1 in the VOLTRL code.

   d. VOLTRL Entry: Call SETMOV, then TAPIO to write a tape mark.

   e. EOV1: TAPIO is called to rewind the tape. Subroutine ASK is called
      to print a message stating:

      MOUNT NEW TAPE. TYPE CU/TR TO CONTINUE/TERMINATE

      and to request the input of 2 characters. The characters are checked
      to be either CU or TR. If neither are entered, the call to ASK is
      repeated. If CU is entered, control goes to the VOLHDR code (after
      the call to SETMOV). If TR is entered, FUNCTION TERMINATED is printed,
      EXDFUN is called, as is ENTPRG, this time with A = 3 (abort entry in
      the calling program). After each of the above calls to TAPIO, if
      there was an error return (Q negative), TAPE I/O ERROR is printed
      followed by an exit to the caller's abort entry.

   f. Subroutine ASK: This subroutine outputs a message, then requests input.
      After calls to ENTSR and SETMOV, the address and length of the message
      to be output are loaded from the locations following the call to ASK.
      The message is transferred to the interrogate buffer and written with
      a call to SUPRW. CONVRS is called to input 2 characters. RTNSR is
      called to pick up the return address and the program exits to the third
      location following the call.

AA5641

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME__ LBL0 - Routine to Write Unlabelled Tapes __ Level 6

INTERFACE

1. ENTRY METHOD

    LBL0 is loaded by calling ENTPRG with A = 0 to 3, corresponding to the
    four entry points, and Q set to select mag. tape function 32 (label 0).

2. INPUT PARAMETERS

    I = ATT base address
    ATT/MTFEXT = function index for return
    ATT/MTFENT = $7FFF when the program is called by TRC

3. EXIT METHOD

    Call to ENTPRG to load the next overlay with Q loaded with the contents
    of ATT/MTFEXT.

4. OUTPUT PARAMETERS OR CONDITIONS

    I = ATT base address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ZERO ($22)
        ONEBIT ($23)
        SST ($47)

    b. SST

        STEMP (20)
        ENTSR ($91)
        SUPRW ($97)
        TRNSR ($98)
        TAPIO ($9A)
        ENTPRG ($9E)
        EXDFUN ($9F)

    c. ATT

        IRB (27)
        MTCA (46)
        MIFENT (46)
        MTFEXT (47)
        USRTMP (60)

    d. CONSTANTS

        FW (3)
        MO (4)

PAGE NO. __399__

AA5641

PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME ___LBLO - Routine to Write Unlabelled Tapes___ ___Level 6___

2. PROGRAM DEPENDENCIES

    ENTSR   (ENEXSR) - Std routine for saving return addresses
    RTNSR   (ENEXSR) - recovers return address from stack
    SUPRW   (PIO)   - prints output message
    TAPIO   (PIO)   - used to rewind and write tape marks
    ENTPRG   - called to load the next overlay
    EXDFUN (ENTPRG) - release the current program
    CONVRS (PIO)   - requests input of a 2 character message
    PRTMSG (SUPER) - prints standard supervisor messages
    SETMOV (ENEXSR) - allows program to move

3. DATA AREAS USED

    ATT - Active terminal table
    IRB - Interrogate buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $81_{16} = 129_{10}$ words.

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Not reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME____LBL1 - Write Standard ANSI Tape Labels_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    LBL1 is a standard Cyberdata tape label program.

    There are four entry points: VOLHDR, FILHDR, FILTRL, and VOLTRL.
    VOLHDR asks for parameter input if required, writes a VOL1 label, and
    continues to FILHDR. FILHDR also asks for parameter input if necessary;
    writes a HDR1 label; writes a HDR2 label if requested; and writes a
    tape mark.

    FILTRL writes an EOF1 label; an EOF2 label, if HDR2 label was written
    at the beginning of the file; and a tape mark.

    VOLTRL writes an EOV1 label and two tape marks, rewinds the tape, and
    requests instructions for continuing on another reel or terminating
    the function. If the continue option is selected, control goes to the
    VOLHDR code to label the new reel. If an end-of-tape condition is
    detected during any of these operations, control goes to the VOLTRL
    code.

    A number of internal subroutines are used to handle the input and editing
    of parameters and the building and writing of the labels.

2. DETAILED DESCRIPTION

    a. VOLHDR: SETMOV is called to allow the program to move.

        EOD1/FILSEC is cleared, as is ATT/ENTPT.

        VOL2 - EOD1/FILSEC is incremented; the tape is rewound with a motion
        call to TAPIO, and MTCA/MTF + 3/12 is tested. If 0, control goes
        to STAN: standard volume header parameters are taken.

        IF MTCA/MTF + 3/12 = 1 the parameters are requested to be input.
        First "VOL HDR 1" is printed (using the subroutine ASK). Then the
        volume serial number and accessibility are requested to be input.
        Subroutine ASK prints the question and waits for the response for each
        parameter. Then ALPCHK is called to check if the response was valid.
        If it wasn't, the question is repeated. If the response was valid the
        response characters are moved from the interrogate buffer to the user
        area in EOD1. If the response is "*E", Q is minus on return from
        ALPCHK and control goes to STAN. For the user I.D, ASK is called
        followed by ALPCHK. In this case the response is not moved to the
        EOD1 buffer.

If A is positive on return from ALPCHK, and Q is greater than
zero, then the end of the interrogate buffer (IRB) is blanked
out (with a call to BLANKS) and the owner ID is moved (with a call
to MOVE) to the position in the IRB corresponding to the position
of the owner ID in the VOL1 label. (character 38 - 51). The words
that would hold the access code in the IRB are blanked, and control
goes to OTH.

STAN - BLANKS is called to clear the entire IRB.
OTH - "VOL1" is placed in the beginning of the IRB.
The volume serial number and access code are moved from the EOD1 user
area to the IRB. WRTLBL is called to write the label. Control goes to
FILCON in the FILHDR code.

b.   FILHDR: SETMOV is called; EOD1/FILSEC and ATT/ENTPT are set to 1
FILCON - based on MTCA/MTF + 3/12, ATT/DEF will be negative if parameters
to be used are default values, positive if new values are to be entered.
If ATT/DEF is negative, control goes to STANF.

"FILE HDR 1" is printed, followed by calls to ASK and ALPCHK to enter
and validate file ID, and set ID. ASK and CHKDAT are called to enter
and validate creation date and expiration date. In each case, if the
parameter is entered incorrectly, it must be reentered. If "*E" is
entered, control passes to STANF, as all following parameters are
defaulted.

STANF - COM1 is called to build the label in the IRB, based on values
in EOD1. "HDR1" is placed at the beginning of the IRB.

The block count parameter is set to zero, and the label is written with
a call to WRTLBL.

If end-of-tape was detected while writing the label, ATT/ENTPT is com-
plemented. If ATT/ENTPT is 2 or 3, then this pass through FILHDR
occurred when LBL1 had been called to write a file or volume trailer.
Therefore HDR2 will be written only if it had previously been written:
control goes to NOTOUT.

If ATT/DEF is positive go to REDHD.

NOTOUT - If a HDR2 has been written on previous files then go to H2,
otherwise go to CC5.

REDMD - request response to "HDR2=". If Y is entered, control goes to
H2. If N is entered, control goes to CC5. If default (carriage return)
is entered, then if there was a previous HDR2, go to H2, otherwise go to
CC5 (any other entry returns control to REDHD).

H2 - subroutine HDR2 is called to build the label in the IRB. "HDR2" is placed in the beginning of the IRB and WRTLBL is called to write the label. If end-of-tape was detected, and ATT/ENTPT isn't negative already it is complemented.

CC5 - TMARK is called to write a tape mark. If end-of-tape was detected or ATT/ENTPT is negative "END OF TAPE" is printed with a call to PRTMSG and control goes to EOV. Otherwise, if ATT/ENTPT = 2 control goes to EOF. For normal file header and volume label calls, the program releases itself (with EXDFUN) and exits to ENTPRG with Q = (ATT/MTFEXT).

c.   FILTRL: SETMOV is called and ATT/ENTPT is set to 2.

EOF - the value in ATT/BLKCNT is moved to EOD1/TOTBLK + 3, COM1 is called to build the label in the IRB, and TMARK is called to write a tape mark. If end-of-tape was detected, then if LBL1 was not called by the tape recovery program (TRC), (ATT/MTFENT = $7FFF), jump to BCK2.

BLKSUB is called to convert the block count to ASCII and store it in the IRB. "EOF1" is placed at the beginning of the IRB and WRTLBL is called. If end-of-tape is detected and if TRC was not the caller, jump to BCK2. If HDR2 had been written at the beginning of the file, EOF2 must be written at the end. Subroutine HDR2 is called to build the label in the IRB, "EOF2" is stored at the start of the IRB, and WRTLBL is called. If no end-of-tape, or end-of-tape and the program was called from TRC jump to AA2.

BCK2 - print "END OF TAPE", back space one record and test for tape mark or end-of-tape. Records are backspaced until a tape mark is found; control goes to EOV11.

AA2 - a tape mark is written. If end-of-tape and not a call from TRC go to BCK2. Otherwise, call EXDFUN and return to the caller with a call to ENTPRG.

d.   VOLTRL: SETMOV is called.

EOV - call COM1 to build the label.

Set ATT/ENTPT to 3.

EOV11 - write a tape mark.

EOV1 - edit the block count and store "EOV1" at the start of the IRB. Call WRTLBL to write the label. Write two tape marks. Rewind the tape, then call ASK to print a message asking for "CU" or "TR" to be entered.

If "CU" is entered, control goes to VOL2.

If "TR" is entered, "FUNCTION TERMINATED" is printed and control goes back to the caller's fourth entry point (A = 3 when calling ENTPRG).

e. Subroutines: There are fifteen subroutines in LBL1. They are described in the order that they appear in the program.

1) BLANKS

On entry, A = relative address in IRB.
          Q = number of words to be blanked - 1
A section of words in the IRB is filled with ASCII blanks.

2) ASK

A message is written and, optionally, a response is received. The following parameters follow the call to ASK:
The (absolute or relative) address of the message to be printed; the message length; and a flag indicating if a response is required or not.

ENTSR and SETMOV are called. The message address is calculated and the messageis moved to the IRB with a call to MOVREC. The message is written with a call to SUPRW. If a response is required, CONVRS is called to input it. If the input device is not a keystation, the last word following the message is set to $FFFF. RTNSR is called to recover the return address. Control goes back to the fourth location after the initial call.

3) ALPCHK - on entry, Q = number of characters in the field to be processed; A = relative address in the EOD1 user area.

SUBDEF is called. If Q is not greater than zero or A is negative on return, the program exits with Q = ATT/DEF.

GETALP is called until an error or end-of-record is detected. If an error is detected, the program exits with A negative. If too many chraracters were entered, the program also exits A negative.

If less than the maximum number of characters was entered, the record is extended with blanks up to the maximum. MOVE is called to move the record from the IRB to the user area in the EOD1 buffer. The program then returns with ATT/DEF in Q and A = 0.

4) CHKDAT - an entered date is checked for validity and moved to the EOD1 buffer if accepted. On entry, A = relative address in EOD1.

SUBDEF is called to check for default entry. If Q is not greater than zero, the program returns to the caller. The entry is checked to be 5 digits followed by end-of-record. If not, the program exits with A negative. The last three digits are checked to be less than 367. If so, the full number is moved to the EOD1 buffer.

5) GETCH - On entry, the local CHRFR is the displacement in characters from the base address in SST/STEMP + 2. The character requested is loaded to A and CHRFR is incremented.

6) MOVE - On entry, Q = target address in IRB and A = source address. Following the call is the number of characters to be moved. If this parameter is negative, start with a shift of one character; if A = 0, the source address is the beginning of the IRB. A loop is set up with successive calls to GETCH, followed by storing of the loaded word to the IRB.

7) COM1 - This routine build the label in the IRB for HRD1, EOF1, and EOV1. The last 33 words of the IRB are filled with blanks. EOD1/FILSEC + 3 and EOD1/FN + 3 are converted to ASCII and stored in the IRB. Generation number and version are set to standard values The file sequence number and ID are moved to the IRB. The creation data is built: if the value is nonzero in EOD1 it is moved to the IRB. Otherwise, the current Julian date is used. If the expiration data was not given "99366" is entered. The accessibility code is moved to the IRB and the program exits.

8) LLSQLS - Converts the upper two BCD digits in A to ASCII and stores them in the address in the local TEMPC1.

9) CNVRT - Converts the binary number in A to ASCII and stores the result in the IRB at the address in Q.

BD is called to convert from binary to BCD. If A is negative on return, it is set to $9999. LLSQLS is called twice.

10) HDR2 - The label contents for HDR2 and EOF2 label is built: the block length (in words) is extracted from the MTF. If zero, 564 is used. The number of words is converted to ASCII number of characters and stored in the IRB. The record length is extracted from the MTF. If this is zero, the block length is used as the record length. The record length is converted to ASCII and stored in the IRB. The reserve area in the label is set to blanks. The buffer offset based on the blocking mode and the record format are stored in the IRB. Control returns to the caller.

11) CHKTRC - Compares ATT/MTFENT with $7FFF.

12) WRTLBL - The output code is isolated from the MTF and TAPIO is called to performed a coded formatted write. If an error occurred, the program exits to the caller of LBL1 after printing "TAPE I/O ERROR" (this is the exit taken for all tape I/O errors throughout the module). If there is no error the program returns with end-of-tape status in A.

13) TMARK - TAPIO is called to write a tape mark.

14) BLKSUB - The block count in EOD1 is converted to ASCII and stored in the IRB.

15) SUBDEF - GETALP is called to analyze the first characters in the IRB. If the first character is an end-of-record ($FF), return with Q = 0. If the first two characters are *E followed by $FF, exit with Q less than 0. Otherwise, exit with Q greater than zero.

INTERFACE

1. ENTRY METHOD

   LBL1 is called by ENTPRG. Execution starts at VOLHDR, FILHDR, FILTRL or VOLTRL, depending on the value of A when ENTPRG was called.

2. INPUT PARAMETERS

   (I) = ATT base address
   ATT/IRB = address of an interrogate buffer
   ATT/EODADR = address of EOD1 buffer
   ATT/MTFEXT = caller's function index
   ATT/MTFENT = $7FFF if caller is TRC

3. EXIT METHOD

   ENTPRG is called with Q = (ATT/MTFEXT)
   A = 0 for normal exits, A = 3 for abort exits

4. OUTPUT PARAMETERS OR CONDITIONS

   (I) = ATT base address

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      TEN ($46)
      SST ($47)

b. SST

   STEMP (20)
   ENTSR ($91)
   EXTSR ($92)
   SUPRW ($97)
   RTNSR ($98)
   GETBIN ($99)
   TAPIO ($9A)
   GETALP ($9B)
   ENTPRG ($9E)
   EXDFUN ($9F)

c. ATT

   IRB (27)
   SVCA (36)
   BCOUNT (SVCA +5)
   MTCA (SVCA + 10)
   SUPINP (36)

d. MTCA

   MTFENT (0 + MTCA)
   MTFEXT (1 + MTCA)
   EODADR (2 + MTCA)
   MTF (3 + MTCA)
   SCRCH (12 + MTCA)
   DEF (SCRCH)
   ENTPT (SCRCH + 1)
   USRTMP (14 + MTCA)

3. EXIT METHOD

   ENTPRG is called with Q = (ATT/MTFEXT)
   A = 0 for normal exits, A = 3 for abort exits

4. OUTPUT PARAMETERS OR CONDITIONS

   (I) = ATT base address

AA5641

DOCUMENT CLASS CYBERDATA IMS

PROGRAM NAME LBL1 - Write Standard ANSI Tape Labels   Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        TEN ($46)
        SST ($47)

    b. SST

        STEMP (20)
        ENTSR ($91)
        EXTSR ($92)
        SUPRW ($97)
        RTNSR ($98)
        GETBIN ($99)
        TAPIO ($9A)
        GETALP ($9B)
        ENTPRG ($9E)
        EXDFUN ($9F)

    c. ATT

        IRB (27)
        SVCA (36)
        BCOUNT (SVCA + 5)
        MTCA (SVCA + 10)
        SUPINP (36)

    d. MTCA

        MTFENT (0 + MTCA)
        MTFEXT (1 + MTCA)
        EODADR (2 + MTCA)
        MTF (3 + MTCA)
        SCRCH (12 + MTCA)
        DEF (SCRCH)
        ENTPT (SCRCH + 1)
        USRTMP (14 + MTCA)

e. EOD1

    TOTBLK (2)
    FN (4)
    FBLK (9)
    USRA (16)

f. Volume Label

    <u>in IRB</u>   <u>In EOD1</u>

    VS (2)   USRVS (0 + USRA)
    ACS (5)   USRACS (3 + USRA)
    OID (18)   FILSEC (22 + USRA)

g. File Label

    <u>in IRB</u>   <u>In EOD1</u>

    FID (2)   USRFID (USRA + 4)
    SID (10)   USRSID (USRA + 13)
    FSN (13)
    CRD (20)   USRCRD (USRA + 16)
    EXD (23)   USREXD (USRA + 19)
    ACSHD (26)
    BLKCNT (27)
                                    USRH2 (USRACS)

h. HDR2
    RECFMT (2)
    BLKLNG (2)
    RECLNG (5)
    OFFSET (25)
    RES (8)

i. EST

    JULDAT (40)

j. CONSTANTS

    FW (3)
    MO (4)
    CFW (19)
    TR ($5452)
    CU ($4355)

AA5641   PRINTED IN THE USA

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   ENTSR   (ENEXSR) - common routine to save return address
   EXTSR   (ENEXSR) - common exit, to last entry in address stack
   RTNSR   (ENEXSR) - recover last entry in address stack
   SUPRW   (PIO)   - called to output messages to the supervisor
   TAPIO   (PIO)   - called to manipulate the tape
   GETALP (SCANV) - checks character in IRB to be alpha
   GETBIN (SCANV) - converts the days in the year to binary
   ENTPRG   - transfers control to the next overlay
   EXDFUN (ENTPRG) - releases the current program
   SETMOV (ENEXSR) - allows program to move in the FRP
   PRTMSO (SUPER) - prints standard supervisor messages
   BD   (ARITH) - converts block count from binary to BCD.
   MOVREC (PIO)   - moves messages to the IRB
   CONVRS (PIO)   - performs input in response to output messages
   AYERTO (SYSDAT) - year in ASCII (last 2 digits)

4. DATA AREAS USED

   ATT (including SVCA, MTCA)
   MTF
   EOD1

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $3B8_{16} = 952_{10}$ words.

2. RELOCATION TYPE

   Run anywhere.

3. ENTRANCY CLASSIFICATION

   Not reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___LBL1 - Write Standard ANSI Tape Labels_____Level 6_

ADDITIONAL INFORMATION

1. For a description of the label formats see CDC Programming Standard
   1.87.002 Magnetic Tape Labels for Information Interchange.

2. Table of entered parameters

Name	Length	Description	Default
Serial No.	6	Field 3 of VOL 1	999999
Accessibility	1	Field 11 of VOL 1	Space
Owner ID	14	Field 7 of VOL 1	Spaces
File ID	17	Field 3 of HDR1 and EOF1	Spaces
Set ID	6	Field 4 of HDR1, EOF1	Serial No.
Creation Date	6	Field 9 of HDR1 and EOF1	System Julian Date
Expiration Date	6	Field 10 of HDR1 and EOF1	99366
HDR2	1	N-No HDR2, Y - Yes	N

AA5641

PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME___LOD1 - LOD Command First Overlay Module_____ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    For the LOD,J command, this module analyzes the command, saves the command
    parameters in the ATT, positions the magnetic tape to the appropriate
    file, (according to the file number parameter) and invokes the second
    overlay (LOD2).

    For the LOD,J command, this module only invokes the second overlay
    (LODFMT).

    LOD,J format:

        LOD,J, job, start batch, finish batch, file no., batch modes

    job - job name, the default is all jobs

    start batch - number of the first batch to be transferred (loaded),
                the default value is 1.

    finish batch - number of the last batch to be loaded, the default
                value is 999.

    file no. - initial tape file number (every batch is a file), the default
            value is 1.

    batch modes - mode of batches to be loaded.

            W - written batches
            C - completed batches or waiting to be verified.
            F - frozen batches

    The default is W,E, F i.e. written, completed and frozen batches will
    be loaded.

2. DETAILED DESCRIPTION

    Step 1 (LOD1)   - Enable function termination by SX or CANCEL key, set
                         the default value of the finish batch and call SETMOV.

    Step 2 (PARM1)   - Check the first parameter. If it is F go to EXITF to
                         release LOD1 and invoke LODF, if it is J go to next
                         step, else go to ERR1 (INVALID) PARAMETER error exit).
                         If there are no more parameters go to EOR (Step 8).

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___LOD1 - LOD Command First Overlay Module_____Level 06___

Step 3 (PAR)	- Check if there is a job name. Parameter, if not go to PARM4 (Step 6). Get the job name and put it in the ATT. If it is the last parameter go to EOR (Step 8).
Step 4 (PARM2)	- Get the start batch number and put it in the ATT. If it is invalid go to ERR3 (INVALID BATCH NUMBER). If it is the last parameter go to EOR (Step 8).
Step 5 (PARM3)	- Get the finish batch number and put it in the ATT. If it is invalid go to ERR3. Check if finish batch start batch, if not go to ERR3. If there are no more parameters, go to EOR (Step 8).
Step 6 (PARM4)	- Get the file number parameter, if it is invalid go to ERR1. If there are no more parameters go to EOR (Step 8).
Step 7 (PARM5)	- Get the batch modes. If there is an invalid or duplicate mode symbol, go to ERR1-
Step 8 (EOR)	- If the job name is not the default option (binary zeroes) check if it is in the LJD, if not go to ERR2 (INVALID JOB NAME).
Step 9 (OPENTP)	- Set the default values of the batch modes which do not appear within the command parameters. Put the file number minus 1 in LTEMP, rewind the tape and skip LTEMP files forward.
Step 10 (EXIT)	- Release LOD1 and call LOD2.

INTERFACE

1. ENTRY METHOD

   Standard supervisor entry.

2. INPUT PARAMETERS

   LOD parameters received in the interrogate buffer.

3. EXIT METHOD

   Releasing LOD1 by EXDFUN, and invoking LOD2 by ENTPRG. If there were errors in the command, exit is to REJSUP with an appropriate message.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME__ LOD1 - LOD Command First Overlay Module __ Level 06

4. OUTPUT PARAMETERS

LOD parameters in the ATT command area.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. SST EQUATES

        SST ($47)
        TEMP1 (20)
        TEMP2 (21)

    b. ATT EQUATES

        TRMNT (23)
        MTCA (46)
        ATEMP (73)
        JN (MTCA)   - job name
        STRBCH (MTCA+3) - start batch
        ENDBCH (MTCA+4) - finish batch
        BMODES (MTCA+5) - batch modes
        FILEN (MTCA+6) - file number
        LTEMP (MTCA+7) - temporary counter

    c. GENERAL EQUATES

        MO (4)   - PIO motion request
        C ($43)   - C ASCII
        F ($46)   - F ASCII
        W ($57)   - W ASCII

2. MACROS USED

    None

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ LOD1 - LOD Command First Overlay Module _____Level 06___

3. PROGRAM DEPENDENCIES

a. ENTRY POINTS

LOD1

b. EXTERNAL LIST

SETMOV
GETLJD

c. TRANSFER VECTOR ROUTINES

ENTPRG
EXTPRG
GETBIN
GETALP
GETCOM
TAPIO
REJSUP

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

242 words.

2. RELOCATION TYPE

Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___LOD2 - Second Overlay of LOD,J Command_____Level 06___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The input for this program is the parameters received from the first
    overlay in the ATT communication area, and the mag tape which contains
    for each recorded batch, a batch header record and the "picture" of the
    batch as it was recorded on the disk when the DMP.T wrote it on the tape.

    The program reads the tape, selects the batches required by the LOD.J
    command, and writes them on the selected disks. The batches are written
    on the disk exactly as they are recorded on the tape, with the exception
    that the batch disk number and all the disk addresses (in the tracks
    table, ATT, and records) are converted to the new disk addresses.

    The program checks that the job name of every new batch is legal and
    that there is no duplicate batch numbers. It checks also that all the
    formats of the new records are in the system.

    At the beginning of a batch loading, its mode is changed to ENTRY and its
    record counter is set to zero so that when power fail occurs during the
    loading of a batch, it is deleted automatically in the next cyberdata
    restart. The batch mode and the record counter are corrected just before
    the completion of the batch loading.

2. DETAILED DESCRIPTION

    These are the main steps of the program:

    a. (LOD2) - Check if the program counter is greater than 1. If it is,
        go to REJSUP with error code 69 (FUNCTION BUSY).

    b. (BEGIN) - Call SETMOV.

    c. (GETRB) - Get TRB buffer for 6 sectors.

    d. (STARTB) - Clear the start flag SFLAG, which denotes that the
        tape is located at the beginning of the next batch.

    e. (NXTBCH) - Check SX or cancel key. If entered, release the TRB
        buffer and terminate the function by exit to REJSUP with error
        code 39.

f. If SFLAG is not zero, skip forward one file, i.e. locate the tape
    to the beginning of the next batch.

g. (REDHDR) - Set SFLAG and read 16 words of the next block. If it is
    and EOF block, then go to step ai (COMPLT).

h. (CHKHDR) - Check if the b-ock, read in g, is a DUMP BATCH HEADER
    or DUMP EOV TRAILER; if not exit to REJSUP with error code 70
    (WRON TAPE). If DUMP header, go to step j and if DUMP EOV trailer
    go to next step.

i. (EØV) - Type: END OF TAPE - MOUNT NEXT TAPE AND TYPE CU. Read the
    operator response. If it is 'cu' rewind the new tape, and return to
    step d. If it is not 'cu', return to the beginning of this step.

j. (CHKJOB) - Get the job name at the next batch from the batch header
    and check if it is a new job, i.e. not equal to the job name of the
    previous batch. If it is the same job, go to step m (OPENB).

k. (NEWJOB) - Check if the new job is requested, if not return to step e.

l. (GETJOB) - Get the new job Parameters from the LJD table. If the job
    name is not found type the message: JOB jjjjjj NOT IN SYSTEM, where
    jjjjjj is the job name, and exit to REJSUP.

m. (OPENB) - Put the batch number in the program messages which require it.

    Check if the new batch is requested, i.e. the batch mode is matched
    with the modes in the command (W.C.F.). If not go to step e.

n. (RED1) - Read the next block from the tape (the first block of the
    batch, after the header) into the TRB, buffer and save the old
    tracks table (first 90 words) in OLDTRK.

o. (GETDSK) - Select the disk for the batch to be loaded, and get a new
    track for it. If all disks are full, type DISKS FULL, and exit to
    REJSUP.

p. (SETFTB) - Update the following ATT entries, of the batch: FTOB,
    JIDX, LUN. Save the record count (RCD), set it to zero and write the
    first 2 sectors of the batch to the disk.

DOCUMENT CLASS ___ CYBERDATA

PROGRAM NAME ___ LOD2 - Second Overlay of LOD,J Command ___ Level 06

g. (WRTABT) - Write the batch entry to ABT, with batch mode ENTRY. If
    the ABT contains entry of another batch, of the same job, with the
    same number, then type: bbb jjjj ALREADY IN THE SYSTEM and go to
    step bi. (DLTBCH).

    NOTE: In case of Power fail, a half-loaded batch will be deleted
    automatically by CLR, at system start-up time.

r. (GTRK) - Get all the tracks for the batch, and save them in the first
    TRB sector. If the disk is full, release the received tracks and
    return to step o to try another disk.

s. (REC1) - Clear the Pointers of track index (TRKIDX) and set the
    next disk address (NXTDA) to the first track address.

t. (NXTREC) - Check if the header of the next record is in the TRB
    buffer, if not go to the next step. If it is, translate the record
    addresses by converting the old track addresses to the next track,
    addresses (the track address in word j of the old tracks table
    OLDTRK is converted to the track address in word j of the new tracks
    table, where j = 1,2,3 . . . . 90) and return to NXTREC.

u. (ENDBLK) - Write the data from the TRB buffer to the next disk address
    (NXTDA). If there is not enough space on the current track, increase
    the track index (TRKIDX) and get the next track from the new tracks
    table.

v. (READB) - Read next block from the tape into the TRB buffer. If
    tape I/Ø error occurs, then type: TAPE I/Ø ERROR - PREVIOUS BATCH
    SKIPPED, and go to step bi (DLTBCH). If the batch was not finished,
    then return to step x (NXTREC).

w. (CLOSE B) - Clear the start flag (SFLAG) and go to FMTCHK.

x. (FMTCHK) - Read the loaded batch and check the record chaining and
    the validity of the format numbers in the records. If the check
    is finished with error, then type a message and go to step bi (DLTBCH).

y. (CLOSE C) - Read the first two sectors of the new batch (tracks table
    and ATT image), update the disk addresses in the ATT image and the
    records counter (R=D). Rewrite these two sectors.

z. (UPABT) - Update the batch mode in the ABT table and go to step d. (STRTB) to load the next batch.

ai. (COMPLT) - Rewind the tape and exit to SV (if the first block is EOF type: WRONG TAPE).

bi. (DLTBCH) - Release the tracks and the ABT entry, allocated to the new batch, and return to step e (NXTBCH) to get the next batch.

INTERFACE

1. ENTRY METHOD

   LOD2 is called from LOD1 by:

       LDQ = N$80A
       CLR   A
       RTJ   (ENTPRG)

2. INPUT PARAMETERS

   RQJN - requested job name (ATT(46))
   BMODES - requested batch modes (ATT(S1)).

3. EXIT METHOD

   a. ERROR EXIT

      JMP - (REJSUP)

   b. NORMAL EXIT

      JMP - (SV)

4. OUTPUT PARAMETERS

   The following is a list of the messages which may be printed by LOD2:

   a. WRONG TAPE

   b. TAPE I/Ø ERROR

   c. TAPE I/Ø ERROR
      PREVIOUS BATCH SKIPPED

   d. FUNCTION TERMINATED

   e. bbbb, jjjjjj
      where bbbb is a batch number and jjjjjj is a job name.

f. bbbb, jjjjjj ALREADY IN SYSTEM

g. JOB jjjjjj NOT IN SYSTEM

h. END OF TAPE - MOUNT NEXT TAPE AND TYPE CU

i. DISKS FULL

j. FORMAT nnnn NOT IN SYSTEM

k. FORMAT nnnn INVALID FOR THIS BATCH

l. ABT FULL

m. JOB MAXIMUM RECORD LENGTH TOO SHORT

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. SST and EST equates:

        SST ($47)
        TEMP1 (2)
        TEMP2 (21)
        TEMP3 (22)
        TEMP4 (23)
        DSKST (29)
        ENTSR ($91)
        EXTSR ($92)
        SUPRW ($97)
        TAPIØ ($9A)
        SPILL ($A1)
        WRTSEC ($A8)
        RELFOR ($AD)
        CAM ($AE)
        WAIT ($B0)
        SV ($B1)
        REJSUP ($B2)
        CLCDWA ($90)
        GFCORE ($9D)
        REDSEC ($A7)
        READN ($A6)
        RTNSR ($98).

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME__LOD2 - Second Overlay of LOD,J Command__ Level 06

   b. ATT Equates

      FTOB (1)
      JIDX (2)
      CMODE (2)
      RCNT (4)
      TRB (6)
      FBA (7)
      RELBA (8)
      CURDA (9)
      NDA (11)
      FMTNØ (13)
      BRCNT (15)
      FMTOP (16)
      BMDA (17)
      SW1 (19)
      ABTIDX (22)
      TRMNT (23)
      DOCFMT (23)
      OLDF1 (24)
      BFMT (25)
      OLDF2 (26)
      IRB (27)
      SRB (28)
      FMTERR (30)
      RCDRD (31)
      RCD (32)
      BATCH (35)
      ATEMP1 (73)
      ATEMP2 (74)
      ATEMP3 (75)
      ATEMP4 (76)

   c. SV. Communication Area Equates

      COMA (46)
      RQJN (COMA)
      STRBCH (COMA+3)
      ENDBCH (COMMA+4)
      BMODES (COMA+5)
      FFLEN (COMA+6)
      LTEMP (COMA+7)
      SFLAG (COMA+8)
      OLDMOD (COMA+9)
      OLDJN (COMA+10)
      PRVDA (COMA+13)
      PRVFMT (COMA+15)
      MAXRL (COMA+16)

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME__ LOD2 - Second Overlay of LOD,J Command   Level 06

      d. General Equates

         BLOCK (4)
         BR (8)
         FR (2)
         FW (3)
         MO (4)

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    EXTERNAL LIST

    SETMOV
    GETLJD
    G1TRT
    LINKTR
    RELTRT
    MOVREC
    BD
    DB
    PRTMSG

4. DATA AREAS AND SYSTEM TABLES USED

    SST
    EST
    ATT
    LJD
    ABT
    FMT
    FBA
    TRT
    DTL

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE

    1312 ($520) words
    Buffers: up to $600

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Not reentrant (locked).

DOCUMENT CLASS ___LODFMT_____

PROGRAM NAME_____Load Formats From Tape To Disk_____

PROGRAM FUNCTION

1. ENTRY POINTS

   Two entry points 1. LOD - LODHDR
                     2. LODF2 - LODHDR

2. DETAILED DESCRIPTION

   Functional Description

   The entered command and format numbers are checked not to contain any input
   and logic errors. For example, format numbers are checked to be increasing
   numbers not greater than the highest format number existing in system. Tape
   is checked to have the right header, DMPF containing date and last block size
   used for formats buffer. If tape is not identified as the proper one, then
   a WRONG TAPE message is sent to supervisor.

   Request is made to CAM for two buffers: 34 words for the tape directory
   formats buffer, according to last printed on tape up to 16 sectors length.
   When received, directory and formats are read to the suitable buffers.
   Directory is checked to contain the requested formats numbers if both requested
   numbers are smaller than those on the tape - the conclusion is that requested
   formats are not on tape and program exits to -SV- if both numbers are greater
   than the requested one, advance is made to the next directory. When a format
   is found in the directory, the program calculates the distance (in words)
   from the beginning of the formats buffer to the beginning of the specified
   format address is saved. The program calls another disk resident program -
   WRTFMT - which writes the format on disk. When format is written, WRTFMT
   calls again the LODFMT program (the second entry point) and the latter checks
   whether the format loaded to disk was the last requested. If not last re-
   quested, it searches the directory further on. Otherwise, a message is printed
   indicating operation is completed and the program exits to -SV-.

INTERFACE

1. ENTRY METHOD

   The supervisor enters the following command:

$$
\text{LOD,F} \left\{ \begin{array}{l} \text{,nnnn(1)} \\ \text{,nnnn(1),nnnn(2)} \\ \text{all} \end{array} \right\} \text{CR}
$$

DOCUMENT CLASS ___LODFMT___

PROGRAM NAME___Load Formats From Tape To Disk___

where: F   - formats load.
        nnnn(1) - first format to be loaded from tape to disk.
        nnnn(2) - last format to be loaded from tape to disk. When last
                  format number is not specified all formats from nnnn(1)
                  to the highest format number on tape will be loaded from
                  tape to disk.
        When no format number is specified, all formats on tape will be loaded.

2. INPUT CONDITIONS

Requested format numbers nnnn(1) and nnnn(2) must be positive integers (not
zero). nnnn(2) greater or equal to nnnn(1) - last format number, can be
greater or equal to the first format number. Neither nnnn(1) nor nnnn(2)
can be greater than the highest format number existing in the system.

3. EXIT METHODS

The program can be terminated by one of the following methods:

Error exit - if an error is detected in the input parameters, in the logic
structure of the request, or if tape is not identified as the right one,
program exits to REJSUP with suitable error message.

Return to caller - when no errors are detected and operation is completed,
program exits to -SV- with a completion message.

4. OUTPUT PARAMETERS

PROGRAM'S ERROR MESSAGES

a. INVALID COMMAND FORMAT - when comma is not following commands name or
   CR is missing at the end of the entered command.

b. INVALID FORMAT NUMBER - when requested format numbers do not fit the
   description in section 2 of INTERFACE.

c. TAPE I/O ERROR - when something is damaged in the tape operation (read,
   write, advance or rewind).

d. WRONG TAPE - when tape is not identified as the requested one.

e. END OF TAPE.

f. MEMORY FULL - RETRY - when CAM fails to get the requested word-size
   buffers.

g. FUNCTION TERMINATED - if MI and SX are pressed during program, execution
   function is terminated.

333

DOCUMENT CLASS ___LODFMT___

PROGRAM NAME___Load Formats From Tape To Disk___

During WRTFMT execution there are several error messages:

a. INVALID RECORD LENGTH - when tapes format record length is greater than the systems format record length.

b. DISK FULL - when format cannot be loaded to disk, lack of space.

c. FMTLNK TABLE FULL - FMTLNK is full and no more formats can be loaded into system.

COMPLETION MESSAGES

a. REQUESTED FORMATS NOT ON TAPE.

b. LAST FORMAT LOADED TO DISK NNNN
   WHERE: NNNN - IS FORMAT NUMBER

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. Low Core Equates

      LPMASK ($2)
      NZERO ($12)
      ONEBIT ($23)
      ZROBIT ($33)
      ZERO ($22)
      TEN ($46)

   b. SST Equates

      SST ($47)
      STEMP ($5B)
      EXTSST (0)
      MAXFMT (24)
      GETBIN ($99)
      GETALP ($9B)
      REJSUP ($B2)
      SV ($B1)
      CAM ($AE)
      WAIT ($B0)
      MAXWT (100)
      RELFOR ($AD)
      CLCDWA ($90)
      WRTDWA ($AA)
      ENTSR ($91)
      EXTSR ($92)

```
             ENTPRG ($9E)
             EXDFUN ($9F)
             TAPIO ($9A)
             MO (4)
             BR (8)
             FW (3)
             SUPRW ($97)

         c. ATT Equates

             PI (21)
             CANC (23)
             IRB (27)
             ATEMP (73)

         d. Communication Area Equates

             SUPCOM (36)
             SC (SUPCOM+10)
             CALRPI (SUPCOM+6)
             FMTBAS (SC+0) - Base address of formats buffer for WRTFMT
             FMTP1 (SC+2)  - First format on current directory
             FMTP2 (SC+3)  - Last format on current directory to tape
             FLAG (SC+4)
             LSTBLK (SC+5) - Big buffer's size in words
             BIG (SC+6)    - Big buffer's backwards pointer
             RELBIG (SC+7) - Relative address in big buffers
             DIR (SC+8)    - Backwards pointer to directory buffer
             RELDIR (SC+9) - Relative address in the directory buffer
             FMT1 (SC+10)  - First requested format no.
             FMT2 (SC+11)  - Last requested format no.
             FMTL (SC+12)  - Last format no. loaded to disk
             BUFSIZ (SC+13)- Big buffer's size (in words)
             FLAG2 (SC+14) - Error flag, contains error code
```

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   Entry points

   ADC   LOD-LODHDR
   ADC   LODFZ-LODHDR

   External list

   SETMOV - Enter in ENEXSR.
            Enables moving of disk resident program.

DOCUMENT CLASS ____LODFMT_____

PROGRAM NAME_____Load Formats From Tape To Disk_____

Routines called via transfer vector

GETALP ($9B) - Entry in SCANV.
                Retrieve the comma following command.

GETBIN (99) - Entry in SCANV.
                Retrieve the formats numbers.

REJSUP ($B2) - Entry in SUPER.
                Exit when input command error occurs or when tape fails.

TAPIO ($9A) - Entry in PIO.
                Performs tape motion reads and writes (from and to specified locations).

SUPRW ($97) - Entry in PIO.
                Called to read or write from or to suitable logical unit.

RELFOR ($AD) - Entry in GFCORE.
                Release the requested format.

SV ($B1)   - Entry in SUPER.
                Exit when command is accomplished.

CAM ($AE)   - Entry in CAM.
                Request core space for program buffers.

WAIT ($B0)   - Entry in ENEXSR.
                Used to delay functions for core cycle.

ENTPRG ($9E) - Entry in ENTPRG.
                Disk resident programs are assessed via ENTPRG, used to call WRTFMT.

EXDFUN ($9F) - Entry in ENTPRG.
                Called to release a disk resident program, releases LODFMT when WRTFMT is entered and vice versa.

4. DATA AREAS AND SYSTEM TABLES USED

LODFMT - None
WRTFMT - Uses the following tables:
        1. LFD
        2. FMT
        3. FLK

5. INTERNAL SUBROUTINES

CHKTAP - checks tape after any tape's operation: read, write, advance or rewind.

            Input parameters: after coming from TAPIO.
                             Register A - contains tape's status.

BLANKS - fills the interrogate buffer with blanks from the end of the message till the end of the buffer, and writes it on tape.

    Input parameters: Register A - Blanks code.
                            Register Q - Serial No. of the first word to be blank.

SERCOR - requests core for the program buffers.

    Input parameters: Register Q - No. of requested words including header.

IONOF   - sets and clears I/O bits.

    Input parameters: Register Q - Backwards pointer of the buffer.

BINASC - converts binary numbers to ASCII characters and stores them in STEMP.

    Input parameters: Register Q - The number to be converted.
                            STEMP   - Place to be stored after conversion.

TRSMSG - transfers requested message to the interrogate buffer.

    Input parameters: Register Q - Pointer to message address.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENT

   $15A_{16}$ = 346 words

2. RELOCATION TYPE

   Disk resident and run anywhere.

3. ENTRANCY CLASSIFICATION

   Re-entrant partially locked:
   WRTFMT is a locked program.

337

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS _____ CYBERDATA _____

PROGRAM NAME_____ LPR - Load Program _____ Level 06 _____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    LPR is a supervisor function used to load new programs (or change existing
    program) on disk, and add (or update) the corresponding entry in PRD. (The
    program to be loaded is assumed to be in the scratch area of the system disk
    when it has been placed by punch absolute file procedure of Libedt.).

    There are two command formats:

    a. LPR,NNN,D

    b. LPR,NNN,D,S

    where:

           NNN = directory index

             D = directory name (miscellaneous, key, interrogate, supervisor,
                 mag-tape, own-code)

                 D = T (mag tape function) ",S" must be entered where S is
                 subdirectory name (MTF own code, tape labelling, code conversion)

    LPR command may be entered only from the TTY.

2. DETAILED DESCRIPTION

    A syntactic check is made on the command. In case of erroneous syntax
    return is made to sup. mode with a suitable message.

    The directory index (program no.) is checked whether it is inside limits
    (in case of own code it is checked against Max. own code number).
    In case of error return is made to sup. mode.

    Next, the address of the corresponding entry in PRD is calculated (stored
    in PTRDA) CAM is invoked to get a buffer for a sector. In case there is
    no core available return is made to sup. mode.

    The first sector of the program is read and the header is checked. In case
    of error return is made to sup. mode.

    The status of the program to be loaded is checked. It may be new (7FFF in
    entry in PRD), existing (nonzero entry in PRD) or being processed (e.g.,
    being read into core; zero entry in PRD). If it is being processed WAIT
    is invoked. If it is an existing program, its length is found (first word
    of header) and checked against the length of the new program.

If the "new" program is shorter than the "old" one, it is written over
the "old" program on disk. A check is made to see if the "old" program
is on SQZ working area. If it is, the new program is written in the
area the program was before power fail.

The "new" program is loaded, sector by sector, on disk.

In case the program to be loaded is new or is longer than the existing
program, it is written in the next available space on disk. If there
is not enough space, return is made to sup. mode.

Where the program existed, it is checked if it is in core. If it is in core,
the user count is checked. If it is zero, the program is moved from core.
The corresponding entries in PRD and PRD on disk are updated. (In the case
of a new program or an old program without enough space, the next available
disk address entry should be updated).

The scratch area is released (zero is written in the "length" word in the
header). The buffer in core is released also before return is made to
supervisor mode.

Internal Routines

GETIND – Finds whether a program is in core or not. (Checks PRD control
block).

On entry – ATEMP,I contains absolute index of program (not relative
to its directory but to first program).

On exit – A = 0 program not in core.
A = 1 program in core.

READIT – Reads a sector.

On entry – READCA is address of the buffer in core.

READDA is address of sector on disk.

On exit – READDA is incremented by 1.

WRWORD – Updates an entry in PRD on disk (one word).

On entry – A contains word displacement from beginning of PRD.
CORE contains address of the buffer the updating is taken from.

DOCUMENT CLASS _____CYBERDATA_____

PROGRAM NAME_____LPR – Load Program_____Level 06_____

INTERFACE

1. ENTRY METHOD

   LPR is disk resident. It is called via the ENTPRG routine.

2. INPUT PARAMETERS

   See command format (general description).

3. EXIT METHOD

   a. completion exit (to SV)

       JMP-   (SV)

   b. error exit (via REJSUP)

       JMP-   (REJSUP)

4. OUTPUT PARAMETERS

   New programs on disk, with starting sector number patched into PRD table.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

       ZROBIT   ($23)

       NZERO   ($12)

       ONEBIT   ($23)

       LPMASK   ($2)

       ZERO   ($22)

       REJSUP   ($B2)

       SV   ($B1)

       WRTSEC   ($A3)

       REDSEC   ($A7)

AA5641

DOCUMENT CLASS ____CYBERDATA_____

PROGRAM NAME____ LPR – Load Program _____Level 06_____

	GETBIN	($99)
	GETALP	($9B)
	CAM	($AE)
	CLCDWA	($90)
	WRTDWA	($AA)
	SUPRW	($97)
	WAIT	($B0)
	STRTRY	($AF)
b.	ATT,SST	
	ATEMP1	(73)
	ATEMP2	(74)
	ATEMP3	(75)
	ATEMP4	(76)
	ATEMP5	(77)
	SST	($47)
	STEMP1	(20)
	STEMP2	(21)
	STEMP3	(22)
	STEMP4	(24)
	STEMP	(20)
	PRDA	(3)
	MAXOCD	(4)
	PRDDA	(10)

AA5641

PRINTED IN THE USA

    c. GENERAL

        FW   (3)

        FLAG   (46)

        MXTAV   (47)

        PTRDA   (48)

        TEMPDA   (49)

        VALUE   (50)

        AREA   (51)

        M   ($4D)

        K   ($4B)

        II   ($49)

        S   ($53)

        T   ($54)

        O   ($4F)

        DSKAD   (3)

        C   ($43)

        L   ($4C)

2. MACROS USED

   None

3. PROGRAM DEPENDENCIES

   SV

   REJSUP

   SUPRW

   WRTSEC

REDSEC

GETBIN

GETCOM

GETALP

WAIT

STRTRY

CAM

CLCDWA

WRTDWA

4. AREAS AND SYSTEM TABLES USED

SST

ATT

EST

PRD (external SIPRDE)

SQZ area (external SIDSQZ)

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$154_{16} = 340_{10}$ words of program.
Buffer of 99 words.

2. RELOCATION TYPE

Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

Not reentrant.

AA5641

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME_____ MATER - Open or Close Batch on Tape   Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    Mater has four entry points:

    ●   WBTEXT closes the current batch as follows:

        Writes a file mark followed by an EOD trailer.

        Clears tape Recovery Area for this batch.

        Displays tape block count.

        Releases all buffer areas allocated for this batch.

        a. NXTBCH opens the next batch on the output tape as follows:

            Displays Tape Recovery Area Number, if it is the first batch
            of the current command.

            Updates the recovery area (check Point) to the current position
            of the tape.

            Displays Batch and File numbers.

            Displays reject message if error condition exists.

            Positions the tape for first record of batch.

            Calls file label processing routine, if necessary.

            Calls appropriate tape processing routine.

        b. Is not used.

        c. ERREXT updates the Active Batch Table to restore the mode of the
            current job and releases all buffer areas allocated for this batch
            before returning to supervisor mode.

2. DETAILED DESCRIPTION

    a. NXTBCH

        1) (NXTBCH). Call SETIO to set ATT/FUNIDX/15 = 1
            (Return address in Exits stack is a relative address).

            ●   Call SETIO to set I/O bit in EOD trailer buffer header.

            ●   Call CLCDWA to calculate disk word address of tape
                recovery area.

            ●   Call WRTDWA to write tape recovery area to disk.

            ●   Call CLRIO to clear I/O bit in EOD trailer buffer header.

2) (START1). Wait until MATER is not in use by other WBT commands.

- If ATT/MTFENT ≠ 0 (not in first batch), go to step 3.

- Call ASCII to convert Tape Recovery Area Number from binary to ASCII code.

- Call SUPRW to issue the message:

  WBT STARTED, RECOVERY AREA nn

  where nn is the tape Recovery Area Number.

- Go to step 4.

3) (PRTBBB). Call ASCII to convert Batch Number (the number of the batch just written) to ASCII.

- Call ASCII to convert File Number to ASCII.

- Call SUPRW to issue the message:

  BBB, FILE NNN

  Where BBB = Batch Number

  and NNN = File Number

- Call SPILSR to update the Active Batch Table (ABT) to set the mode of the current batch to written mode.

- Increment ATT/STRBAX (Current batch number by one).

4) (NEXTB1). If (ATT/STRBAX) > (ATT/ENDBAX) (last batch), go to step 17.

- Call BD to convert the next batch number to Packed Decimal; save the decimal number in ATT/BATCH.

- If ATT/TRNNT/14 = 1 (cancel key was entered), go to step b2.

- Call SPILSR to locate the ABT entry for the current batch number.

- If there is no such entry, get the entry of the smallest batch number of the current job, which is greater than the current batch number. If there are no more entries in the current job, go to step 17.

- Call DB to convert current batch number to binary; save it in ATT/STRBAX. If this represents the last batch in ABT, go to step 17.

- Save job index in ATT/JIDX, batch number is ATT/BATCH, and batch track address is ATT/FTOB.

- If batch is in Entry or Verify mode, go to step 5.

- If batch is in Maintenance mode, go to step 6.

- Set flags in SST/TEMP1 according to condition of batch:

  Bit 0 = 1   Batch written

  Bit 2 = 1   Batch unbalanced

  Bit 4 = 1   Error flagged record in batch

- Move ATT/STATUS/4,2,0 to SST/TEMP2/4,2,0 ('Y' bits of the status Parameters ER, UB and WR, respectively, meaning only batches of type ER, UB, or WR are included).

- Move ATT/STATUS/5,3,1 to SST/TEMP 3/4,2,0 ('N' bits of the status parameters ER, UB, and WR, respectively meaning no batches of type ER, UB, or WR are included).

- If SST/TEMP1/4,2, or 0 does not match SST/TEMP2/4/2/ or 0, respectively, and does match SST/TEMP3/4,2, or 0, respectively go to step 5 (Inconsistent status).

- If EST/DSKST/1u + 4 = 1 (Disk unit down), when 1u = ATT/BATCH/12-13 (Logical unit assigned to batch), call PRTMSG with (A) = 66 + 1u, to issue the message:

  DISK n DESELECTED (n = 1u + 1)
  and go to step 6.

- Call SPILSR to update ABT entry to indicate the batch is in Maintenance mode, and go to step 7.

5) (REJBCH). If ATT/STATUS/15 = 1 (STOP parameter in WBT command) go to step 6.

   - If ATT/STATUS/4,2,0 = 0 (Batches with ER, UB, or WR status are not included,) go to step 6.

   - Otherwise go to step 4.

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___MATER - Open or Close Batch on Tape_____Level 06

6) (STOP1). Call ASCII to convert batch number to ASCII; insert it in reject message.

- Insert batch mode, ER status, and UB status in reject message.

- Call SUPRW to issue the message:

    bbbb   REJECTED, Mode = x, UB = y, ER = z

    where bbbb = batch number

                x = batch mode: E = Entry
                                  V = Verify
                                  W = Written
                                  M = Maintenance

                y = UB status   (Y or N)

                z = ER status   (Y or N)

- If ATT/STATUS/15 = 1 (STOP parameter in WBT command) go to step b2.

- If not, go to step 4.

7) (BSTART). If ATT/MTFENT ≠ 0 (first entry to MATER, from WBT) then go to step 12.

- If ATT/BCOUNT = 0 (Beginning of tape), go to step 11.

- If ATT/MTFNUM ≠ Previous Magnetic Tape Format Number, go to step 11.

- If ATT/MTF + 3/0 - 11 = 0 (No batch grouping, i.e., batches are to be recorded in one file, go to step 8.

- If ATT/JNAME through ATT/JNAME + 2 = Previous Job Name, go to step 11.

- Call CHKBG to check batch grouping.

- If the current batch should be grouped in a new file, go to step 11.

8) (OVRTRL). Set ATT/MTFENT = 0 (Adding to current file).

- If ATT/BCOUNT < Total block count for this reel, go to step b3.

- If ATT/BCOUNT = Total block count for this reel, go to step 9.

- ATT/BCOUNT > Total block count for this reel, call TAPIO to backspace (ATT/BCOUNT - Total Block count) records.

- If **NOT** we backed over a file mark, go to step b3.

AA5641

9) (CLEAR). Move Magnetic Tape Format Number and Job Name from ATT to EOD trailer. Clear Magnetic Tape Format Number and Job Name in ATT.

10) (MONEXT). Set ATT/MTFEXT = MTFMON (Label processor is to return to MTFMON and go to step 13.

11) (HDREXT). Set ATT/MTFENT = 20 (Starting new file) and go to step 9.

12) (CMKEOF). Call CHKBG to check batch grouping.

   ● If current batch is grouped with previous batch, set ATT/MTFENT = 0 (Adding to current file) and go to step 10.

   ● If current batch is not grouped with previous batch, set ATT/MTFENT = 20 (Starting new file).

   ● Set ATT/MTFEXT = LBL + Label number, the ENTPRG a-parameter which calls for a file label processing routine.

13) (BBB). If ATT/MTFEXT = MTFMON, go to step 14.

   ● If not, call EXDFUN to release MATER program area.

   ● Set GATE = 0 (make MATER available for use).

   ● Set ATT/MTFEXT = MTFMON (Label processor is to return to MTFMON.)

   ● Call ENTPRG to load and execute entry point 2 in the designated file label processing routine.

14) (LSTBLK). If ATT/MTFENT = 20 (Starting new file) go to step 15.

   ● If ATT/MTF/14-15 = 3 (Fixed Block Mode), go to step 16.

15) (CALMON). Call EXDFUN to release MATER program area.

   ● Set GATE = 0 (Make MATER available for use).

   ● Call ENTPRG to load and execute entry point 0 in the Magnetic Tape processing routing designated in ATT/MTFEXT.

16) (FIXBLK). If current block is full, or if no more data remains to be blocked, go to step 15.

- Put the number of characters, of the last block, in word 3 of the output buffer.

- Set I/O bit in the output buffer.

- Call TAPIO to backspace the tape one block.

- If an error was detected, go to step b3.

- Call TAPIO to read the last block into the output buffer.

- If an error was detected, go to step b3.

- Call TAPIO to backspace the tape one block.

- If an error was detected, go to step b3.

- If not, go to step 15.

17) (LASTB). Set GATE = 0 (Make MATER available for use by other terminals).

- Set ATT/MTFENT = 30 (Last batch).

- If ATT/BCOUNT = 0 (Beginning of Tape), go to step b1.

- Call EXDFUN to release MATER program area.

- Set ATT/MTFEXT = MTFMON (Label processor is to return to MTFMON).

- Call ENTPRG to load and execute entry point 2 in the designated file label processing routine.

b. WBTEXT

1) (WBTEXT). Call SETIO to set ATT/FUNIDX/15 = 1 (Return address to be stored in Exits stack is relative address).

- If GATE ≠ 0, call WAIT to delay one cycle; repeat this step until GATE = 0.

- Increment GATE by one.

- Call SETPOS to convert block count to ASCII and store the result in the block total message.

- Call TAPIO to write a file mark.

- If an error was detected, go to step c3.

AA5641

- Build the 16-word record:

  EODI ***************************

  in the Interrogate buffer.

- Call TAPIO to write the EOD record on the output tape.

- If any error was detected, go to step c3.

- Call SETIO to set the I/O bit in the End of Data Trailer area header block.

- Call TAPIO to write the End of Data Trailer on the output tape.

- If any error was detected, go to step c3.

- Call CLDWA to calculate disk word address of Magnetic Tape recovery area.

- Call REDDWA to read 4-word recovery area control block.

- Clear the control bit for the tape recovery area associated with the current batch.

- Call WRTDWA to re-write the 4-word recovery area control block.

- Call SUPRW to issue the message:

  TOTAL NUMBER OF BLOCKS NNNNN
  where NNNNN is the tape block count.

- Set GATE = 0 (Make MATER available for use).

- If ATT/MTFENT ≠ 30 (Not on last batch), go to step c1.

- Call RELEAS to release buffers that are assigned:

  End of Data Trailer, Terminal Record Buffer, Input Buffer, Output Buffer, Interrogate Buffer, Format Buffer, Code Conversion Program.

- Exit to SV to process next Supervisor Command.

2) (TERMIN). Call PRTMSG to issue the message FUNCTION TERMINATED and go to step a17.

3) (TIOERR). Set GATE = 0 (Make MATER available for use).

- Call PRTMSG to issue the message TAPE I/O ERROR and continue at step c.

c. ERREXT

    1) (ERREXT). Call SETIO to set ATT/FUNIDX/15 = 1 (Return address to be stored in Exits stack is relative address).

      ● If GATE ≠ 0, call WAIT to delay one cycle; repeat this step until GATE = 0.

      ● Increment GATE by one.

      ● If ATT/JIDX = 0 (no open batch at this terminal) go to step 2,

      ● Call SPILSR to update the Active Batch Table to restore the mode of the current batch.

    2) (OUT). Call RELEAS to release assigned buffers.

      ● Exit to REJSUP with (A) = $FFFF (No error message).

    3) (TIOER1). Call RELEAS to release assigned buffers.●

      ● Exit to REJSUP with (A) = 58 (Issue TAPE I/O ERROR message).

INTERFACE

1. ENTRY METHOD

MATER is a disk-resident program which is called via ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

(I) = Base address of Active Terminal Table (ATT)

3. EXIT METHOD

Normally, NXTBCH calls another disk-resident program, either MTFMON or a label-processing program. In case of error, exit is to REJSUP.

WBTEXT exits to SV normally; to REJSUP in case of error.

ERREXT exits to REJSUP.

(I) = Base address of ATT.

AA5641

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME_____MATER - Open or Close Batch on Tape_____Level 06

ASSEMBLY CHARACTERISTICS

1. EQUATES .

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        CLCDWA ($90)
        ENTSR ($91)
        EXTSR ($92)
        SUPRW ($97)
        TAPIO ($9A)
        ENTPRG ($9E)
        EXDFUN ($9F)
        SPILL ($A1)
        REDDWA ($A9)
        WRTDWA ($AA)
        RELFOR ($AD)
        WAIT ($B0)
        SV ($B1)
        REJSUP ($B2)

    b. SST

        TEMP1 (20)
        TEMP2 (21)
        TEMP3 (22)
        DSKST (29)

    c. ATT

        FTOB (1)
        JIDX (3)
        INBUF (5)
        TRB (6)
        FBA (7)
        FMTNO (13)
        FUNIDX (21)
        ABTIDX (22)
        TRMNT (23)
        IRB (27)

AA5641

DOCUMENT CLASS __CYBERDATA_____

PROGRAM NAME__MATER - Open or Close Batch on Tape_____Level 06

       OUTBUT (28)
       BATCH (35)
       SVCA (36)
       BCOUNT (SVCA + 5)
       MTCA (SVCA + 10)
       MTFENT (MTCA)
       MTFEXT (MTCA + 1)
       EODADR (MTCA + 2)
       MTF (MTCA + 3)
       STRBAX (MTCA + 8)
       ENDBAX (MTCA + 9)
       STATUS (MTCA + 10)
       TRANUM (MTCA + 11)
       JNAME (MTCA + 12)
       MTFNUM (MTCA + 15)
       ATEMP (73)

  d. EOD TRAILER

       TOTBLK (2)
       MTFN (3)
       FN (4)
       JN (5)
       BN (8)
       LBC (12)
       EODLNG (40)

  e. TAPIO REQUEST CODES

       MO (4)
       BFW (11)
       CODFR (18)
       CODFW (19)

  f. ENTPRG REQUEST CODES

       MATER ($0805)
       MTFMOV ($080F)
       LBL ($4820)

2. MACROS USED

   None

3. PROGRAM DEPENDENCIES

| ENTPRG | { | ENTPRG | Enter a disk resident program. |
| | | EXDFUN | Exit disk resident program. |

ENEXSR	{	SETMOV	Enable moving of disk resident program.
		ENTSR	Save return address in Exits stack.
		EXTSR	Exit to last return address in Exits stack.
		WAIT	Schedule delay.

| SPILL | | SPILL | Maintain ABT. |

| GFCORE | | RELFOR | Release format. |

| ARITH | { | BD | Binary-to-decimal conversion. |
| | | DB | Decimal-to-binary conversion. |

PIO	{	CLCDWA	Calculate disk word address.
		SUPRW	Supervisor read-write.
		TAPIO	Tape I/O processor.
		REDDWA	Read disk word address.
		WRTDWA	Write disk word address.

SUPER	{	PRTMSG	Write a message at supervisor console.
		SV	Supervisor command processor.
		.REJSUP	Supervisor command reject routine.

4. DATA AREAS USED

SST	System Status Table
ATT	Active Terminal Table
EOD	End of Data Trailer
TRA	Tape Recovery Area
IRB	Interrogate Buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $311_{16} = 785_{10}$ words.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Not reentrant (closed).

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___MCHAR__ - __Move Character into Self-Scan Terminal Output Buffer_____ Level 0

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

MCHAR enters a character into the data portion of the self-scan TOPT. If
the character being entered does not fit in the existing data area (causes
overflow), the indicator bits on all characters in the data section are
set so that the entire area is redisplayed with the new character at the
right end of the display. A character may be stored in the self-scan TOPT
at any time because an indicator is set for each character that has not been
displayed.

2. DETAILED DESCRIPTION

First MCHAR checks to see if the next character will cause a new overflow
situation, (if next available position is 32). If so, the closed subroutine
GETDEF is called to see which area of the display is currently in use
(idle supervisor, interrogate of data) to reset the next available character
position. The overflow count (TOPT/HDR2/9-13) is set to 1, the overflow
flag is set, and (SETIND) the indicator bit to be displayed is set on all
characters in the data field. (NORMAL) The character is stored in the next
available character position and the next available character position is
incremented. The address of the leftmost character for output is updated
to the latest character address if overflow occurred. Interrupts are inhib-
ited during the manipulation of the data buffer itself. Control is returned
to the caller by indirect jump through the entry cell.

If the addition of this character does not cause a new overflow condition
but an overflow condition was already in existence, a check is made to see
if the new character will be the last one in the buffer. If so, the over-
flow count is set to zero. If not the last character, the overflow count is
incremented by 1. Then control jumps to SETIND (described above).

If the new character does not cause overflow and no overflow condition
existed, control goes to NORMAL (described above).

INTERFACE

1. ENTRY METHOD

Return jump

2. INPUT PARAMETERS OR CONDITIONS

1 = current ATT base address

ATT/CURCH = character to be stored

AA5641

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___MCHAR - Move Character into Self-Scan Terminal Output Buffer___ Level 06

3. EXIT METHOD

   Jump indirect through entry cell

4. OUTPUT PARAMETERS OR CONDITIONS

   1 = current ATT address

   Character is stored in TOPT

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK   (2)

      ONEBIT   ($23)

      ZROBIT   ($33)

      SST   ($47)

   b. ATT

      CURCH   (2)

      TOPTAD   (33)

   c. SST

      CATT   (11)

   d. TOPT

      HDR2   (2)

      HDR3   (3)

      DATBAS   (4)

   e. DISPLAY DATA DEFINITION STARTING COLUMNS

      D3   (0)

      D2   (0)

      D1   (3)

      DO   (14)

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME___ MCHAR - Move Character into Self-Scan Terminal Output Buffer _____ Level 06

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

None

4. DATA AREAS USED

ATT

SST

TOPT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$60_{16} = 109_{10}$ words.

2. RELOCATION TYPE

Loader relocatable.

3. ENTRANCY CLASSIFICATION

Dormant reentrant.

AA5641

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ MDMY - Miscellaneous Dummy Function _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

MDMY is the dummy program for miscellaneous functions.

2. DETAILED DESCRIPTION

MDMY calls MONITR with a return jump, and Q register set to 8006, indicating error report number 6 is requested (nonexistent function called). Upon return from MONITR, MDMY hangs the computer on 18FF.

INTERFACE

1. ENTRY METHOD

MDMY is a disk resident program loaded by ENTPRG, after a miscellaneous type of function has been entered.

ENTPRG checks if the program needed is in core or being loaded into core. If finds out it is neither in core nor being loaded, a check is made to see if the program exists on disk. If not, a request is made to load the miscellaneous dummy function program, MDMY.

2. INPUT PARAMETERS OR CONDITIONS

None.

3. EXIT METHOD

Program makes computer hand on 18FF.

ASSEMBLY CHARACTERISTICS

1. EQUATES

SST
MONITR ($8C)

2. PROGRAM DEPENDENCIES

MONITR generates error report.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ MDMY - Msicellaneous Dummy Function ___ Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    9_{16} words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    Three operating system programs were changed to incorporate the CYBERDATA system requirements:

    a. MINT   - To allow for scheduling the supervisor at any time (SP).
               - To allow abortion of Cyberdata at any time (VX).
               - To allow abortion of supervisor at any time (SX).

    b. SPACE - To allow for dynamic setting of memory (allocatable and unprotected)˙depended upon previously set system requirements.

    c. MIPRO - To allow for Cyberdata system loading (VL).

2. DETAILED DESCRIPTION

    a. MINT

        Since standard OS MINT/MIPRO might be locked (MIB/M/MB nonzero) the modified MINT allows for Cyberdata requests processing even when busy.

        If MINT and MIPRO are not busy, request processing is normal except for 'VX' 'SP' and 'SX' which commands are processed in MINT (not MIPRO). In case the normal processing path is busy, the following conditions are checked:

        a) A second request is not processed.

        b) I/O not in progress.

        c) Cyberdata is active.

        If all the a.m. conditions are met, a secondary READ is issued and the request processing is normal, but only 'VX' 'SP' or 'SX' are legal. Requests processing is done by the 'CKSPVX' subroutine.

        This subroutine performs the following functions:

        1)   'SX' request -

                Set terminate bit on the Active·Terminal Table for the supervisor.

        2)   'SP' request -

                Schedule SV entry point at Cyberdata operation level.

        3)   'VX' request -

                Set total system lock to prevent additional terminals input. Scan all terminals for I/O in progress, and wait for I/O completion. Then clear Cyberdata active flag and release memory occupied by Cyberdata and type 'AB'.

b. SPACE

Restart portion of SPACE was modified to allow for dynamic setting of ALLOCATABLE CORE areas lengths.

Area setting is according to the flags set by 'VLOS'.

c. MIPRO

The following Cyberdata system commands were added:

1) 'VL' - load Cyberdata.

MIPRO will check that Cyberdata activation is not inhibited and not already active.

If loading is permitted VLSTRT, the start up routine will be scheduled.

2) 'VLTP' -

Schedule Cyberdata terminal type change routine.

3) 'VLBG' -

Schedule Cyberdata and Background relation control routine.

4) 'VLOS' -

Schedule start up control routine.

5) 'VLST' -

Schedule Cyberdata start up status print routine.

INTERFACE

1. ENTRY METHOD

a. MINT - Normal, no change from MSOS.

b. SPACE - Normal.

c. MIPRO - Normal.

2. INPUT PARAMETERS

a. MINT - Normal parameter input.

b. SPACE - System flags modified by 'VLOS'.

c. MIPRO - Normal parameter input.

3. EXIT METHOD

    a. MINT

        1)   SX request - JMP to dispatcher.

        2)   SP request - schedule SV entry point.

        3)   VX request - JMP to dispatcher.

    b. SPACE

       Normal.

    c. MIPRO

       Schedule requested processor.

4. OUTPUT PARAMETER

    a. MINT

        1)   'SX' request - Set terminate flag.

        2)   'SP' request - None.

        3)   'VX' request - Type 'AB'.

    b. SPACE

       Changes in LVLSTR.

    c. MIPRO

       None.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. MINT

       CP(3), CPVL(5)
       SV($B1), SST($47)
       ATTL(19, ATTB(10), FRPLL(6), TRMNT(23)
       RQ(65), T(2), SLOCK(1)

    b. SPACE

       SST($47)
       WES1(30)

    c. MIPRO

       SST($47), WES1(30)

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___MINT, SPACE, MIPRO - MSOS Changes_____Level 6

2. CONSTANTS

    a. MINT

       None.

    b. SPACE

       None.

    c. MIPRO

       None.

3. PROGRAM DEPENDENCIES

    a. MINT - MIBVL external.

    b. SPACE - VLOS externals.

       VLA5S
       VLA5
       BGCORE
       LFTOVR
       TOTAVL
       VLSTRT

    c. MIPRO

       VLSTRT external
       VLTYPE
       VLBGOP
       VLOSOP
       VLSTTS
       VLOS
       MIBVL

4. DATA AREAS USED

    a. MINT - Internal buffer

    b. SPACE - None

    c. MIPRO - Internal buffer

AA5641

DOCUMENT CLASS ___CYBERDATA. IMS_____

PROGRAM NAME_____MINT, SPACE, MIPRO - MSOS Changes_____ Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    a. MINT - $19F_{16}$ = 415_{10} words.

    b. SPACE - 208_{16} = 520_{10} words.

    c. MIPRO -   -   word.

2. RELOCATION TYPE

    a. MINT - Absolute

    b. SPACE - Absolute

    c. MIPRO - Run anywhere.

3. ENTRANCY

    a. MINT - Not reentrant

    b. SPACE - Not reentrant

    c. MIPRO - Not reentrant

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___MISKEY - Backspace Character or Display Miscellaneous Message on CRT- Level 06___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   MISKEY has two entry points: CBSCEV and CSKUP.
   CBSCEV backs up one position on the CRT.
   CSKUP displays designated data on the CRT.

2. DETAILED DESCRIPTION

   a. CBSCEV

      1) (CBSCEV). Call ENDSR to save return address.

         o Decrement current screen position in data area or interrogate area by 1.

         o Call CSR to convert cursor position on two 5-bit characters.

         o Insert cursor position characters in CDATA calling sequence.

         o If in data area of screen, call CCOL to display column number.

      2) (BSCDRV). Call CDATA to blank out position that was backed over.

         o Exit to EXDSR.

   b. CSKUP

      1) (CKSUP). Call ENDSR to save return address.

         o Insert Data address in PAR1 of CDATA calling sequence.

         o Construct request definition word in REQ of CDATA calling sequence:

15	14	13	12	11	10	8	7			0
M	P		O	B	0 0 0				N	

         m = 1   modifier bit defines parameter string
         P = 1   One parameter in addition to basic 2 parameters
         B = Byte position (0 = upper byte)
                          (1 = lower byte)
         N = number of characters in message.

         o Insert address of second word of Terminal Record Buffer (word with I/O bit) in PAR2 of CDATA calling sequence.

         o Call CDATA to display message.

         o Exit to EXDSR.

AA5641

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME____MISKEY - Backspace Character or Display Miscellaneous Message on CRT- Level 06

INTERFACE

1. ENTRY METHOD

    a. CBSEV

```
        EXT   CB5CEV
        RTJ   CBSCEV    No parameter
        --Return--
        (I) = Base address of Active Terminal Table (ATT)
```

    b. CSKUP

```
        LDQ   Data Address
        STQ   RQ + 1, 1
        LDA   Byte position + No. of characters
        STA   RQ, I                              bit 15 = byte position
        EXT   CSKUP                                     = 0, upper byte
        RTJ   CSKUP                                     = 1, lower byte
        --Return--
        (I) = Base address of ATT
```

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

```
        LPMASK (2)
        ONEBIT ($23)
        SST ($47)
        CDATA ($8D)
        ENDSR ($8E)
        EXDSR ($8F)
```

    b. SST

```
        TEMP (20)
```

    c. ATT

```
        ATRB (6)
        TOPT (33)
        RQ (65)
```

d. TOPT

    CURSRD (3)
    BLOKS (14)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

VL2CRT	(CDATA)	Display data on CRT
CMISC1	ENDSR	Enter display subroutine
	EXDSR	Exit display subroutine
MISKEY	(CONCSR)	Convert cursor position
HDRNOS	(CCOL)	Display current column number on CRT

4. DATA AREAS USED

SST	System Status Table
ATT	Active Terminal Table
TOPT	Terminal Output Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$30_{16} = 48_{10}$ words.

2. RELOCATION TYPE

MISKEY is core resident, loader relocatable.

3. ENTRANCY CLASSIFICATION

Not reentrant.

AA5641

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME_____MLDSKY - Manual/Automatic Dup/Skip_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    When the key is pressed, the automatic function specified on the
    format specification sheet is not performed, and M is shown on the
    key entry station display. New data may be keyed in this field.
    When the key is pressed again, the M disappears from the display and
    the specified automatic function is performed.

2. DETAILED DESCRIPTION

    MLDSKY checks the current mode. The function is allowed only under
    Entry mode or Read mode (with Pseudo Entry).

    If Pseudo Entry then a check is made for Insert. If Replace Field,
    then current field must be auto-dup - else error (exit via ERROR).
    The function now flips the Dup/Skip indicator.

    *   (SW1 BIT 6 = 1, all auto fields are to be entered manually till
        the next Multilevel Dup/Skip key).

        If not at end of field exit is via CLRSR; else, if end of record, exit
        is via CLRSR; or if auto Dup/Skip bit is now set (SW1 BIT 6 = 1)
        then exit via CLRSR. Otherwise not if Replace Field exit via CLRSR;
        else releases function, and exits to EMM5 (entry point in EMM) to
        perform AUTO.

3. INTERFACE

    a. Entry Method

        MLDSKY is a disk resident function which is invoked by ENTPRG.

    b. Input Parameter

        (I) = Base address of ATT.

    c. Output Parameter

        SW1 BIT 6 (ATT word 19) - (See * Detailed Description).

    d. Exit Method

        Exit is to ERROR if an error is detected; to CLRSR otherwise.
        If not at EOF in ENTRY or INSERT; else to EMM5.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____MLDSKY - Manual/Automatic Dup/Skip_____Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        SST ($47)
        LPMASK (2)
        EXDFUN ($9F)
        ERROR ($96)
        DISPLY ($A0)
        CLRSR ($93)
        ONEBIT ($23)
        ZROBIT ($33)

    B. ATT

        RCNT (4)
        FCNT (3)
        BMODE (15)
        ATRB (6)
        SW1 (19)
        MODE (2)

2. EXTERNALS

    SETMOV
    DMLDS
    FIRSTF
    EMM5
    DFNDTY

3. PROGRAM DEPENDENCIES

    SETMOV
    EMM5 (EMM)
    FIRSTF (EDRENT) - to display updated AID line
    ERROR
    DISPLY
    EXDFUN
    CLRSR

4. DATA AREAS USED

    ATT - Active Terminal Table
    SST - System Status Table

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME ___MLDSKY - Manual/Automatic Dup/Skip___ Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $38_{16} = 56_{10}$ words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ MONITR - Debug Utility___   Level 06

PROGRAM FUNCTION

1. MONITR

    MONITR operates in two modes. In error report mode, it prints one line of
    error information and halts the CPU to allow selective dumping of memory.
    In test mode, it performs checks for corruption of system data and, if an
    error is detected, executes error report mode.

2. TRACE

    TRACE prints a line of information with an error code of 99 (trace only
    code).

3. MONDMP

    MONDMP prints a line of information using the last error code generated
    by MONITR (or TRACE).

DETAILED DESCRIPTION

1. MONITR entry

    a. A, Q, and I register are saved in save cells. A and Q are also saved
        in error save cells. Q is examined to determine whether MONITR is to
        execute error report or test and control is passed to 2 or 3 (below).

2. Error report mode

    a. The current terminal number is obtained from the ATT and combined with
        the error number (input in Q). This information is placed in error
        save Q.

    b. Interrupts are locked out and words 0, 1 of memory are modified so that
        a restart of the CPU goes to 2e.

    c. MONDMP (see 6) is called to dump error save registers, other registers
        and caller address.

    d. If entered for a dummy call, control is passed to step 2e. Otherwise,
        the CPU is halted (a restart will pass control to 2e.

    e. Words 0, 1 of memory are restored to their original contents, the Q,A,I
        registers are restored, interrupts are enabled and MONITR returns to
        the caller.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME __MONITR - Debug Utility__   Level 06

3. Test mode

  a. The internal MONITR list of the 3 most recently serviced terminals is
      updated for the terminal number in the current ATT.

  b. If the character count in the supervisor TIQ is nonzero, Q is set to 7
      and control goes to 3d.

  c. The sum of words 0 - $46 in memory is calculated. If no check sum has
      previously been saved, the sum is saved. The current sum is compared
      with the check sum and, if they are equal, control is passed to 3c.
      If not equal, Q is set to 5 and control passed to 3d.

  d. This is common error code for test mode in MONITR. A is placed in error
      save A and control passes to step 2a (start of error report mode).

  e. The FRP chain is checked for corruption as follows:

      For each area, the area length is added to the starting address to get
      the next area's starting address. If the next area start equals the
      FRP pointer in the SST, control passes to 3f. If the next area start
      is below the FRP pointer (a), Q is set to 4 and control passes to 3d.

  f. The number of terminals in the system, as specified in the SST, if
      checked. If it is less than equal to 64, control passes to 3g. If
      greater than 64 Q is set to 8, A is the number of terminals in the
      SST and control goes to 3d.

  g. If select skip is set, control goes to 2e (MONITR exit point).
      Otherwise the active terminal count is cleared.

  h. If the first ATT indicates an active terminal, the active terminal count
      is incremented. If the first ATT does not specify terminal number 0
      and supervisor terminal, control goes to 3j. Else current terminal
      number is set to 0.

  i. The input unit, list unit, basic input unit, and basic list unit in the
      first ATT are checked. If all are nonzero, control goes to 3l. If any
      is 0, its offset from the ATT start is saved in A (upper byte) and control
      goes to 3k (ATT error).

  j. The first ATT either did not specify terminal 0 or did not indicate a
      supervisor terminal. A is cleared and control goes to 3k.

AA5641

k. Q is set to 10 and control passed to the test error exit (3d). This represents ATT corruption error.

l. Control is passed to 4y, to begin testing other terminals and increment the current terminal number.

4. Test mode, loop to test terminals

    a. The current ATT address is incremented by the ATT length (in SST) to get to the next terminal's ATT.

    b. The terminal number in the ATT is compared to the current terminal number. If not equal, A is set to current terminal number and the control passes to 3k (ATT error).

    c. Verify that the hardware address associated with this terminal's controller (in the EST) is the same as that saved in the terminal's output buffer (pointed to by the ATT). If these do not match, set A = offset to terminal output buffer address in ATT and go to 3k (ATT error).

    d. Examine switch 2 in the ATT. If the bit indicating that the ATT check is to be overridden is set, go to 4x (bypass).

    e. Get the address of the backward pointer in the interrogate parameters buffer from the ATT and get the current mode word from the ATT. If the interrogate mode bit is set and the interrogate buffer address was 0, set A = offset interrogate buffer address and go to 3k (ATT error).

    f. If no interrogate buffer exists or the interrogate buffer's I/O bit is set, go to 4g. Otherwise verify that the interrogate buffer backward link is the same as the address of the ATT cell containing the buffer address. If these do not match, set A = offset in ATT to interrogate buffer address and go to 4k (ATT error).

    g. Examine current mode word in ATT. If terminal is not idle or supervisor go to 4i. Otherwise IDK bit must be set. If IDK mode is not set, set A = offset to CMODE in ATT and go to 3k (ATT error). If interrogate mode is set, go to 4x (process next terminal).

    h. Examine the ATT terminal status word (word 0). If either function or pseudo function lock is set, go to 4x (next terminal). If both are clear, go to 4y (bypass active terminal count increment and proceed to next terminal).

    i. Get the terminal record buffer address from the ATT and verify that its backward pointer equals the address of the ATT cell for the terminal record buffer address. If not equal, set A = offset of TRB from start of ATT and go to 3k (ATT error).

AA5641

j. Get the job index word in the ATT and isolate the job index. If job
    index is 0, set A = ATT offset to job index word and go to 3k (ATT error).

k. Get available words in track from the ATT and verify that is less
    than 601. If greater than 600, set A = offset to available tracks and
    got to 3k (ATT error).

l. Get address of format base address table from the ATT. If the address is
    not 0 (formal table exist), go to 4o. Otherwise examine the ATT terminal
    status word. If either function or pseudo function lock is set, go to 4q.

m. (No format table exists and no function locks are set). If IDK mode is
    not set in the ATT current mode word, go to 4p. Otherwise check ATT
    switch 1 to see if format selection must be performed. If such a request
    has been flagged, go to 4q. Otherwise go to 4p.

n. (A format table exists). Get the base address of the format in the table
    and get the back link from that format. Verify that the back link points
    to the format's entry in the format table. If these are equal, go to 4q.

o. Set A = offset to format table and go to 3k (ATT error).

p. Get ATT current mode. If current mode is READ, go to 4s. Otherwise
    compare current mode with terminal's basic mode (also saved in ATT).
    If current mode is different from basic, go to 4r. If they are the
    same and basic mode is entry, examine that neither pseudo entry nor
    pseudo verify set, if any of these two bits is set, go to 4r. If basic
    mode is verify, examine current mode word to see if pseudo verify bit is
    set. If not set, go to 4s.

q. Set A = ATT offset to current mode word and go to 3k (ATT error).

r. Get current record character count from ATT. If this is negative or
    greater than the maximum record length as specified in the EST, set
    A = offset to current record character count in ATT and go to 3k
    (ATT error).

s. Get total number of records in batch from ATT. If negative, go to 4u.
    If larger or equal number of records in read mode (in ATT) go to 4v.
    If number of records in read mode greater than total number of records
    and ATT basic mode specifies entry mode, set A = ATT offset to number
    records in read mode and go to 3k (ATT error).

t. Set A = ATT offset to total number records in batch and go to 3k
    (ATT error).

AA5641

u. Get last entry/exit level address from ATT stack. If this is nonzero, set A = ATT offset to end of stack and go to 3k (ATT error).

v. Get function index word from ATT and isolate return address type flag. If flag indicates that return address is relative and no function index is specified, set A = ATT offset to function index and go to 3k (ATT error).

w. Increment counter of active terminals.

x. Get total number of terminals in system from SST and compare with current terminal number. If these are not equal, the current terminal number is incremented and control passed to 4a (begin testing for next terminal). If the current terminal is the last, the number of active terminals is checked and if it is nonzero, control is passed to 4z. If there are no active terminals, the FRP start and end addresses in the SST are checked. If these are equal (the buffer should be empty if no terminals are active) control goes to 4z. Otherwise, control goes to 4e (FRB chain error).

y. Control is passed to the monitor exit point 2e.

5. TRACE entry point

a. Save A and Q. Set A = 99 (nonerror type error code) and save in error save Q. Clear MIBVL.

b. Set A = caller address and set in caller cell for MONDMP. Set A = saved A.

c. Call MONDMP (see 6) to print dump header line.

d. A - EXTFLG (=0).

e. If A ≠ 0, call CORDMP (see i) to dump memory.

f. Restore A, Q, increment MIBVL and return to caller.

AA5641

6. MONDMP entry point

    a. Save original A, Q, registers.

    b. Get current terminal number from SST, convert to decimal and save in
        buffer. Get error code from error save Q, convert to decimal and save
        in buffer. Get caller address, convert to ASCII and place in buffer.

    c. Convert original Q,A,M and I to ASCII and place in buffer.

    d. Convert current time (in SST) to ASCII hours, minutes and seconds.
        Get string of the form HH: MM: SS in the buffer.

    e. Print the buffer and wait for completion.

    f. Return to caller.

INTERFACE

1. ENTRY METHOD

    a. MONITR is called via   RTJ   MONITR

    b. TRACE is called via   RTJ   TRACE

    c. MONDMP is called via   RTJ   MONDMP

2. INPUT PARAMETERS

    a. For MONITR

        Q $\geq$ 0 indicates test mode
        Q $<$ 0 indicates error report mode
            Q = $8000 + error number
            A = error relavent information

    b. For TRACE

        There are no input parameters

    c. For MONDMP

        The word after the RTJ instruction is dumped as the caller address.

DOCUMENT CLASS __CYBERDATA_____

PROGRAM NAME___MONITR - Debug Utility_____Level 06

3. EXIT METHOD

    a. MONITR and TRACE exit via JMP through word 0 of the subroutine.

    b. MONDMP increments word 0 of the subroutine by 1 before the JMP.

4. OUTPUT PARAMETERS OR CONDITIONS

    None

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

       LPMASK (2)
       NZERO ($12)
       ZERO ($22)
       ONEBIT ($23)
       TEN ($46)
       SST ($47)

    b. SST

       FRB (6)
       FRBPT (8)
       ATTS (10)
       CATT (11)
       TIQ (15)
       ATTL (19)

    c. EST

       WES (30)
       MAXREC (38)

AA5641

     d. ATT

         FTOB (C1)
         CMODE (2)
         JIDX (3)
         RCNT (4)
         TRB (6)
         AVLTK (8)
         BMOD (15)
         SW1 (19)
         SW2 (20)
         PI (21)
         AINT (27)
         RCDRD (31)
         RCD (32)
         ATOPT (33)
         ENDSTK (85)

2. MACROS USED

    FWRITE

3. PROGRAM DEPENDENCIES

    None

4. DATA AREAS USED

    a. SST   (System Status Table)
    b. EST   (Extended System Table)
    c. ATT   (Active Terminal Table)
    d. FRP   (Format/Record/Program buffer)
    e. TIQ   (Terminal Input Queue)
    f. TRB   (Terminal Record Buffer)
    g. FBA   (Format Base Address Table)
    h. TOPT   (Terminal Output Table)
    i   FMT   (Format Table)
    j. INT   (Interrogate Request Buffer)
    k. SCA   (Supervisor Communication Area in ATT)

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___MONITR - Debug Utility___   Level 06

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $2B2_{16}$ words.

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Dormant reentrant.

AA5641

MONITOR ERRORS

T=XX E=XX P=XXXX Q=XXXX A=XXXX M-XXXX I-XXXX HH:MM:SS

E	T	Q	A	I	DESCRIPTION
01	Terminal No.	Unused	TRB ADR	ATT ADR	Disk Address Loop (Terminal accessing disk left locked)
02	Terminal No.	Unused	Unused	ATT ADR	Stack Full
03	Terminal No.	Unused	Unused	ATT ADR	No Index (EXDFUN)
04	Terminal No.	Unused	Unused	ATT ADR	FRP Chain Corrupted
05	Unused	Unused	Unused	Unused	Low Core Checksum Error
06	Terminal No.	Unused	Unused	ATT ADR	Nonexistent function called (ENTPRG)
07	Terminal No.	Unused	Unused	ATT ADR	Supervisors TIQ Nonzero
08	No. of Terms	Unused	Unused	Unused	Illegal No. of Terminals
09	Terminal No.	Unused	Unused	ATT ADR	No Conversion Table
10	Word/Term	Unused	Unused	ATT ADR	
	(High order 8 bits contain index into ATT low order 8 bits contain terminal number)				error in ATT
11	Terminal No.	Unused	Unused	ATT ADR	No Supervisor When SV Called
12					Unused
13	Terminal No.	Unused	Unused	ATT ADR	Format No. 0 Referenced
99	Unused	Core Location	PRD Index	ATT ARD	Not an Error

(The above printout can be caused with the patch LHC, MONITR, 154/ 580E. Each time a disk resident module is loaded into core the above message is printed giving information on the module loaded.)

CONTROL DATA CORPORATION ○ PSSD ○ SOFTWARE DOCUMENT

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME_____MPC - Main Process Control_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    Main Process Control is the routine that controls the processing of
    each entry in a terminal input queue. The entry point MPC is initially
    scheduled for terminal 0 by the terminal controller interrupt handler
    (VLINT). MPC schedules itself for each subsequent terminal which has
    unprocessed key input, handling one key entry for one terminal each
    time it is scheduled. After the last terminal is processed, the memory
    move module is executed if necessary and if time permits.

    There are three other entry points within MPC. (1) CONVRT is where the
    character and terminal mode checking begins. (2) ENTFUN handles function
    keys by loading and executing the appropriate program module via ENTPRG.
    (3) CHR does character validation for entry, verify, pseudo entry and
    pseudo verify modes.

2. DETAILED DESCRIPTION

    a. MPC: If MPC is scheduled for station 0 (TTY) the internal value is
        saved in the local location TIME, after the program completes to scan
        all the stations it checks whether a new interrupt was entered during
        the processing of this cycle. If an interrupt was entered, MPC
        reschedules itself for station 0, otherwise it exits to dispatcher.

        Before MPC starts to process the data of the station where to it was
        scheduled, it schedules the next station that has any function or
        data to be processed. The memory move module (MMM) is scheduled
        as the maximum station No. plus 1. When MPC is scheduled for MMM
        it pushes up the buffers in the FRP, a buffer at a time, till there
        are no more buffers available to be pushed on. A clock interrupt
        was entered into the system.

        MPC calculates the current ATT address (I) and the current format
        field address (CFFA) and exits to MONITR to check whether the system
        tables and buffers seem to be correct. If everything is all right,
        MPC continues. 'LL' is displayed if the TIQ (Terminal Input Queue)
        is full and the station is processed according to the following
        conditions:

        If the station is function locked, i.e. an I/O is being performed
        exit to DISPCH.

        If the station is pseudo function locked, i.e. the previous function
        was not completed because of any delays, exit to RETRY to continue the
        processing of the previous function.

        If the station is not in interrogate mode but the central supervisor
        wants to send her a message, exit to CLRSR to display 'SM' on the
        station.

If the station is in supervisor lock, exit to DISPCH.

If the station is not locked the oldest entry in the TIQ is extracted
and the queue pointer is updated. If the station was system locked the
lock is reset and the 'LL' is cleared from the screen. The keystrokes
counter is incremented by one.

b. CONVRT : The entry is converted to an ASCII character or a function
code according to keyboard type. If the terminal is in Interrogate
mode, the following things happen based on the key entry:

REL ——————
1. Store terminator 'FF' in interrogate buffer.
2. Clear ATT/CMODE/15 (interrogate mode indicator).
3. Put completion code in header of interrogate buffer.
4. Common subroutine exit.

PRLFLD ——————
1. Clears the characters already entered.
2. Jump to dispatcher.

CNCL ——————
1. If supervisor mode, sets ATT/TRMNT/14 (cancel key
   entered bit).
2. Clear ATT/CMODE/15 (Interrogate mode indicator).
3. Put completion code in header of interrogate buffer.
4. Common subroutine exit.

BSC ——————
1. Backspaces a character.
2. Jump to dispatcher.

Any Other Code
1. If function key put @ in ATT/CURCH/0-7 (current
   character value in ASCII).
2. If waiting for continue or cancel key to to 6.
3. If Supervisor mode, and a function key was entered
   jump to dispatcher.
4. If Supervisor mode, clear interrogate line
   if first character in
5. Display the character saved in ATT/CURCH/0-7.
6. Store the entered key in interrogate buffer.
7. Store terminator 'FF' in interrogate buffer.
8. If the number of characters entered ≠ number needed
   for the function, jump to dispatcher.
9. If the number of characters entered = number needed
   for the function, put overflow completion code in
   interrogate buffer header.
10. Clear ATT/CMODE/15 (Interrogate mode indicator).
11. Common subroutine exit.

If the terminal is not in Interrogate mode, an IDK mode check is made. If
the terminal is in IDK mode, MPC jumps to the IDKCH module.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME__ MPC - Main Process Control   Level 6__

c.   ENTFUN : If the entry was a function key, the appropriate function
module is loaded and executed via the ENTPRG module.

d.   CHR : If the entry was an alphanumeric character, basic ASCII alpha,
numeric and sign range checks are made on the character according to
ATT/DT/13-15 (field data type). If the character passes the field data
type check, ATT/CMODE/12,13 is checked for Pseudo Entry and Pseudo Verify
modes and ATT/CMODE/8-11 is checked for Entry, Verify and Read mode. If
the terminal is in Pseudo Entry or Entry mode, MPC jumps to the EMM
module. If the terminal is in Pseudo Verify or Verify mode, MPC jumps
to the VMM module. If the terminal is in Read mode, an A3 (invalid
character) error is displayed. If the character does not pass the field
data type check, an A1 (data type error -- numeric only/alpha only) is
displayed. If the data type is signed, the character is a sign, and
the terminal mode is Entry or Pseudo Entry, the skip function module is
loaded and executed via ENTPRG.

INTERFACE

1.   ENTRY METHOD

MPC: Scheduled at priority 5.
CONVRT, ENTFUN, CHR: Direct jump.

2.   INPUT PARAMETERS OR CONDITIONS

a.   MPC: Q = current terminal number

b.   CONVRT: I = ATT base address
                   ATTTMP,I = character keyed in

c.   ENTFUN: I = ATT base address
                   CURCH,I = Current function code (ASCII)

d.   CHR: I = ATT base address
                   CURCH, I = Current character value (ASCII)

On return from the code conversion routine ATT/CURCH/0-7 = converted code
     Q=0 if the key entered was a function key,   else
     0= ASCII code of the character.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___MPC - Main Process Control___   Level 6

3. EXIT METHODS

```
JMP - (DISPCH)     Jump to dispatcher
JMP - (EXTSR)      Common subroutine exit
JMP - IDKCH        Jump to IDK mode routine
```

where:

```
Q = 0 if function code
Q = ASCII code of the entered alphanumeric character
RTJ - (ENTPRG)      Load and execute disk resident program
JMP - (ERROR)       Display error at terminal
JMP - VMM           Jump to Verify mode module
JMP - EMM           Jump to Entry mode module
RTJ   CONO29  ⎫
RTJ   CONTTY  ⎬     Convert the entered code to the appropriate
RTJ   CONADD  ⎪     ASCII Value
RTJ   CONTTN  ⎭
ATTTMP,I = character keyed in
```

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

```
LPMAST (2)
LPMSK (2)
NZERO ($12)
ZERO ($22)
ONEBIT ($23)
ZROBIT ($33)
SST ($47)
MONITR ($8C)
EXTSR ($92)
CLRSR ($93)
ERROR ($96)
ENTPRG ($9E)
DISP ($A0)
DISPCH ($EA)
```

    b. SST

```
SLOCK (1)
FRPP (8)
ATTS (10)
ATT (11)
FFA (12)
IBL (14)
TIQ (15)
NTRVAL (18)
ATTLNG (19)
STEMP (20)
```

c. ATT

CMODE (2)
CURCH (2)
PFBA (7)
DT (13)
RFA (14)
SW2 (20)
FUNIDX (21)
TRMNT (23)
KEYDPR (26)
AINT (27)
ECODE (34)
REQNOC (72)
ATTTMP (73)

d. ERROR CODES

CBLNK (0)
CA3 (3)
CA1 (1)
CPP ($1A)

e. CONSTANTS

BSC ($1A)
AT ($40)
RES ($67)
CSKIP ($68)
REL ($6F)
DSPKEY ($73)
RPFLD ($74)
CANCEL ($76)

f. ENTERNALS

CONO 29
CONTTY
CONADD
CONTTN
IDKCH
VMM
EMM
MMM
DBSCEV
DCHAR
DERR
DLOCK
CLRFLD

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME____ MPC - Main Process Control_____ Level 6

2. PROGRAM DEPENDENCIES

CONO29	O29 Keyboard Character Conversion
CONTTY	TTY Keyboard Character Conversion
CONADD	O29 + Adding Machine Character Conversion
CONTTN	TTY + Numeric Block Character Conversion
IDKCH	IDK Check
VMM	Verify Mode Processor
EMM	Entry Mode Processor
CAM (MMM)	Core Allocation (Memory Move Module)
DISPCH	Dispatcher
DISPLY	Display Routines
ENEXSR	Common Entry/Exist Subroutine
EXTSR	
CLRSR	
ENTPRG	Enter a disk resident program
GSE ERROR	Display Error to Terminal
MONITR	Debug Routine

3. DATA AREAS USED

SST	System Status Table
ATT	Active Terminal Table
TIQ	Terminal Input Queue

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $17E = 382_{10}$ words

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Reentrant code.

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

The program provides a method of communication between the central
supervisor and (remote) station operators, thus allowing the supervisor
to direct the action of the operators and the operators to consult the
supervisor. Being in SUP mode, the central supervisor may send messages
to any station - depending on the working mode of the key station.
The reaction is as follows:

a. In Idle - The screen is cleared and the first 10 characters of the
message are displayed.

b. In SUP - (But not in 'MSG' mode) the first 10 characters of the message
are displayed following the - SV - on the first line.

c. In ENT, VER and READ - Error code 'SM' is displayed and the first
24 characters of the message are recorded on disk, allowing
further display (only the last 8 messages are saved).

To hold a conversation the operator should enter the sup. command MSG.
If the supervisor is Idle, "MSGnn" (nn = The operator's station number)
is printed on her TTY. To enter the conversational mode the supervisor
should enter MSGnn, followed by her acknowledgement that she is ready.
Conversational mode is terminated by the message END. The use of CNCL
by the operator or MI SX by the supervisor will abort the conversation.

2. DETAILED DESCRIPTION

After the program enables itself to be moved (SETMOV) and sets the flag
enabling "CANCEL" to be entered, the command SYNTAX is checked according
to the caller's identity. If the supervisor called the command format
should be 'MSGxx'; if a station is called, command should be 'MSG'.

a. Supervisor Called

Validity of station number is checked. (The 'MSG' command is illegal
for self-scan).

Next the station mode is checked. If supervisor mode, GO TO SUPR, else
if ENT or VER mode, GO TO ENTVER; else GO TO IDLE.

AA5641

(IDLE). Lock TTY, transfer control to station clear station's screen, transfer control back to TTY, unlock TTY. GO TO TENCH.

(SUPR). If in MSG mode GO TO MSG03.

(TENCH). The message is checked for possible errors and corrected (SET) and trailing blacks are inserted (SUBR). Five words (10 characters) are moved from int. buff. to the TOPT; The MSG flag is set in word 15 of TOPT and the request is treaded at top of CRT Thread. Return is made to SV.

(ENTVER). The address the message is to be written to is calculated according to the station number and the MSG counter (cyclic counter, module 8). Trailing blanks are inserted and message is written. 'SM' flag is set for MPC. Return is made to SV.

b. Operator Called

It is checked if TTY is busy. If it is busy the program exits to WAIT for one cycle thus enabling the operator to cancel the request in case the supervisor is busy for a long time. In case CANCEL is entered, the screen is cleared, the MSG flag is cleared and return is made to REJSUP. If cancel was not entered a retry is made. If the TTY is not busy (NOBUSY), the message MSGxx (xx = station number) is inserted in the output buffer, and an FWRITE request is built in the SC area. The function lock is set on TTY, and the request is performed. On completion function lock of TTY is cleared and station is set to 'MSG' mode.

The principle behind this part is that the station being processed hangs on WAIT until the other station schedules a message for it (FLAG2 = 0). The exchange of messages is done by inserting the message in the interrogate buffer of the sending station and exchanging buffers (pointers) with the receiving station. (MSG03) (This part is common to the processing of a station calling or superv. calling.) The station being processed is set to 'MSG' mode and the indicator that no message was scheduled for that station (FLAG2) is set. If the station being processed is not TTY, GO TO CHECK; if TTY the message in the Int. buffer is checked and corrected (SET), GO TO CHANGE.

(CHECK). The station being processed waits until the other station answers, or cancel is entered on the station being processed. If cancel was entered it is checked if the other station is still on-line (by ON-LINE), if it is on line cancel is entered for it and in both cases the function terminates (after cleaning screen and MSG flags). If the other station answers (FLAG2 is cleared), the int. buffer is displayed (printed). (Note that int. buffers were switched in CHANGE). A check for END is done. If the message was END, the function terminates and exit to SV is made. If not, the message in its own int. buffer is read, checked and displayed (if CRT on int. line). Then the check if the other station is still on line is done. If it is not, the function terminates; otherwise GO TO CHANGE.

(CHANGE). FLAG2 for the other station is set (it means a message was scheduled for it) and is reset for its own. Pointer and backward pointers to int. buffers are exchanged. If the message (its own message) was END, cut-off conversation (checked afterward by ON-LINE), if not GO TO check.

Internal Subroutines

SET

Checks the message in Int. buffer and corrects non-ASCII characters by inserting $40 instead.

SUBR

Fills in the message with trailing blanks (to a total length of 24 chars).

ONLINE

Checks if the other station is still in MSG mode, if not (conversation was cut-off) it returns zero in Q, otherwise returns in Q ATT address of the other station.

PRINT

Displays the int. buffer, if called by a CRT station checks for end of page condition and clears screen if met before displaying line.

INTERFACE

1. ENTRY METHOD

    Invoked by DSKSUP (via ENTPRG)

2. EXIT METHOD

    Via SV - normal exit

    Via REJSUP - abnormal exit (function aborted).

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___MSG___   Level 06

ASSEMBLY CHARACTERISTICS

1. EQUATES

    CLCDWA ($90)
    ENTSR ($91)
    EXTSR ($92)
    SUPRW ($97)
    GETBIN ($99)
    GETALP ($9B)
    DISP ($A0)
    WRTDWA ($AA)
    WAIT ($B0)
    SV ($B1)
    REJSUP ($B2)
    SST ($47)
    SLOCK (1)
    TOPCRT (4)
    BATT (10)
    ATTB (10
    CATT (11)
    ATTLNG (19)
    STEMP (20)
    CMODE (2)
    SW2 (20)
    FUNIDX (21)
    TØIT (33)
    CANC (23)
    AINT (27)
    MSGP (78)
    ATEMPT (73)
    RQ (65)
    C (1)
    T (2)
    LU (3)
    N (4)
    S (5)
    SC (36 = 10)
    ØWNNØ (SC+2)
    BUFF (SC+2)
    FLAG2 (SC+5)
    LPMASK (2)
    NZERØ ($12)
    ZERØ ($22)
    ØNEBIT ($23)
    ZRØBIT ($33)
    TEN ($46)
    DCR ($44FF)
    MØT (5)
    FR (2)
    FW (3)

2. EXTERNALS

   SETMOV
   SIBMSG
   CLRALL
   SGTDIS
   REGEN
   CCKEND

3. DATA AREAS USED

   SST
   ATT
   TOPT
   IRB

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $IAC_{16} = 428_{10}$ program length.

2. RELOCATION TYPE

   Loader relocatable.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ MTC - Magnetic Tape Control Command Processor___ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    MTC interprets a Supervisor Magnetic Tape Control Command (MTC), and issues the corresponding motion request to the TAPIO module.

    MTC commands are:

MTC, RW		Rewind
MTC, RU		Rewind Unload
MTC, AR, M	Advance M records
MTC, BR, M	Backspace M files
MTC, AF, M	Advance M files
MTC, BF, M	Backspace M files
MTC, TM, M	Write M tape marks
MTC, DN, M	Select recording density M

2. DETAILED DESCRIPTION

    a. (MTC). Call SETMOV to enable MTC to be moved, if necessary.

        ● Call GETALP to get next character from the input string. MTC expects a comma at this point.

        ● If end of record (EOR) was detected, set (A) = 8 and exit to REJSUP to issue INCOMPLETE COMMAND message and process next supervisor command.

        ● If any character other than a comma is next, go to step g.

    b. (PARAM1). If comma is next character, compare motion command with entries in MCODE table.

        ● If no match is found, go to step g.

    c. (SETMC). When motion command is located in MCODE table, Q holds the motion command index. Save it in TAPIO request.

        ● Set ATT/REP = 0 (Repetition count).

        ● Set SST/TEMP1 = 6 (Current position in input string).

        ● If motion command index is 3 (Rewind) or 4 (Rewind Unload), no repetition count is required, so go to step d.

        ● Call GETALP to get next character from input string. If it is not a comma, go to step g.

        ● Call GETBIN to extract the repetition count from the input string and convert it to binary. If an error is detected in the input string, go to step g. Save the binary number in ATT/REP. If not, save the binary number in ATT/REP, and go to step e.

d. (CHKLST). Call GETALP to get the next character from the input string. Check if the character is a space or end of record sign, if not, go to step g. If it is a space, call GETCOM and check that the remaining characters are also spaces; if not, go to step g.

e. If motion command request code (MOPARM) is zero (select density), go to step h.

f. (MORQ)

    ●   Call TAPIO to perform the tape motion request.

    ●   If error code returned (Q negative), exit to REJSUP to issue TAPE I/O ERROR message, and proceed to the next SV command.

    ●   Reduce the repetitions number (REP) leg 1 and check it. If it is negative go to SV exit. Check if SX command or cancel key was keyed and terminate the function, by exit to REJSUP, if they were.

    ●   Return to MORQ to repeat the motion request execution.

g. (ERRI). Set (A) = 3, and exit to REJSUP to issue INVALID PARAMETER message and process next supervisor command.

h. (DEN). Set parameter value in TAPIO request to denote recording density as follows:

    parameter = 1, density = 800 BPZ
    parameter = 2, density = 556 BPZ
    parameter = 3, density = 200 BPZ
    parameter = 4, density =1600 BPZ

    o   If the requested density is one of 200, 556, 800, or 1600, go to step f. to issue TAPIO request to set density.

    o   If not, go to step g.

INTERFACE

1. ENTRY METHOD

    MTC is a disk-resident module.
    Entry is via a call to ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

    (I) = Base address of Active Terminal Table (ATT).

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____MTC - Magnetic Tape Control Command Processor_____Level 6

3. EXIT METHOD

Normal exit is to SV.
Error exit is to REJSUP.

4. OUTPUT PARAMETERS OR CONDITIONS

(1) = Base address of ATT.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        ONEBIT ($23)
        SST ($47)
        GETBIN ($99)
        TAPIO ($9A)
        GETALP ($9B)
        SV ($B1)
        REJSUP ($B2).
        GETCOM ($9C)

    b. SST

        TEMP1 (20)

    c. ATT

        CANC (23) - cancel or SX
        IRB (27) - INTERROGATE BUFF ADD.
        MTCA (46) - MAG TAPE COMM. AREA
        MC (MTCA) - MOTION CODE
        REP (MTCA + 1) - REPETITION NUMBER

    d. TAPIO REQUEST PARAMETER

        MO (4)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

SCANV	{ GETBIN	Get a number from inptu and convert it to binary.
	GETALP	Get an alpha character from input string.
PIO	(TAPIO)	Tape I/O request processor.
SUPER	SV	Supervisor command processor.
	REJSUP	Display error message and process supervisor command.
ENEXSR	(SETMOV)	Enable moving of disk resident program.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __MTC - Magnetic Tape Control Command Processor__   Level 6

4. DATA AREAS USED

    SST   System Status Table
    ATT   Active Terminal Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $73_{16} = 115_{10}$ words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___MTFMON - Mag Tape Format Monitor___ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

MTFMON is a disk resident program which supervises mag tape writing.
It connects between: MATER, RSQ, MTF own codes and labels programs. The
following diagram shows the interaction:

2. DETAILED DESCRIPTION

The function of MTFMON is dependent upon its selected entry and on the
value of MTFENT at each of these entries:

When control is transferred to ENTRY1, MTFENT is checked as follows:

If MTFENT=0, control transfers to BBATCH (beginning of batch) to read the
first batch input block.

If MTFENT=1, control transfers to EBATCH (end of batch) where MATER is
called, to get the next batch.

If MTFENT=2, control transfers to EVOL (end of volume) where the appropriate
label program is called to process the volume trailer label.

If MTFENT=10, control transfers to MTOCD (mag tape own code) to call the
MTF own code, if it is not zero. The entry to MTFMON with MTFENT=10 is
made from RSQ.

If MTFENT=20, control transfers to NEWFIL (new file) where the appropriate
label program is called to process the file header label.

If MTFENT=30, control transfers to MTREXT (MATER exit) to call MATER, after
the last batch has been written to the tape.

If MTFENT=-1, control transfers to GETREC to get the next record from the
input buffer. When the current record is the last record of the batch, the
output buffer (OUTBUF) is written to the tape, MTFENT is set to 1 and the
MTF own code is invoked. The MTF -wn code returns to MTFMON at ENTRY1 where
MTFENT=1. If the MTF own code is zero then control returns directly to
ENTRY1.

When the current record is not the last record of the batch, the disk address
of the next record is picked up, from the current record header, and
translated relative to the beginning of the current input block (the data
of INBUF).

If the next record is contained in the input buffer, its relative address
(in the buffer) is replaced by the current record relative address (in RELBA)
and it becomes the current record. If the next record is not contained in
the buffer, a new block which begins with the current record, is read into
the buffer and the next record becomes the current. After the current
record address was updated, the record is checked if it requires RSQ
reformatting. If RSQ is required, MTFENT is set to 10 and RSQ is invoked.

If RSQ is not required then the current record is transferred from INBUF to the record buffer (TRB) and control transfers to MTOCD, to invoke the MTF own code before putting the record in the output buffer (MTFENT=-1). Usually the control returns from the MTF own code to ENTRY2, with MTFENT=-1, to put the current record in the output buffer (OUTBUF).

ENTRY2 is the entry point where the processing of output blocking is begun. The records are grouped in the output buffer according to the MTF parameters (see EMT command in Cyberdata System Reference Manual).

Before each record is transferred to the output buffer, it is converted to the appropriate code (according to the recording code in the MTF). When the recording code is ASCII, then code conversion is not required. When it is BCD (for 7 track only) the conversion is done by the mag tape driver and when the recording code is EBCDIC the conversion is done by invoking EBCDIC program, which remains in memory until the end of the current WBT command. When the output buffer is filled and prepared for writing the program control is transferred to ENTRY3, to write it on the tape. ENTRY3 may be used also by the MTF own code, when non-standard blocking mode is used. In this case (when the MTF own code returns to MTFMON through ENTRY3), control returns to the own code immediately after writing the output buffer on the tape. If END OF TAPE is sensed during the writing, then MTFENT is set to 2 and the MTF own code (if not zero) is invoked to perform special end of volume procedures. Control returns from the own code to ENTRY1, with MTFENT=2, and proceeds to EVOL to process the standard end of volume procedure. If tape I/O error occurs during the I/O operation, then the message TAPE I/O ERROR is printed and control transfers to ENTRY4, where TRC is invoked in order to close the tape correctly at the end of the previous batch.

INTERFACE

1. ENTRY METHOD

    The entry to MTFMON is done by ENTPRG, to one of MTFMON four entries.

2. INPUT PARAMETERS

    a. MTFENT   -   MTFMON, and its disk routines, entry parameter which
                     designate the following states on entry to MTFMON:

                     MTFENT=0   - beginning of batch (from MATER or label).
                     MTFENT=1   - end of batch (from own code).
                     MTFENT=2   - end of volume (from own code).
                     MTFENT=10   - return from RSQ.
                     MTFENT=20   - new file (from MATER or label).
                     MTFENT=30   - last batch completed (from MATER).
                     MTFENT=-1   - get next record (from own code).

    b. MTFEXT   - MTFMON, and its disk routines, return function index for ENTPRG.

    c. EODADR   - EOD buffer address.

    d. MTF   - current MTF (see also EMT IMS).

    e. STRBAX   - current batch number.

    f. ENDBAX   - finish batch number.

    g. STATUS   - WBT Recovery Area number.

    h. TRANUM   - Tape Recovery Area number.

       NOTE: For more details, especially for MTF and STATUS, see MTCA in Cyberdata Software User's Guide.

3. EXIT METHOD

The general scheme of exit from MTFMON is:

```
RTJ - (EXDFUN) - release MTFMON.
LDQ - =XFUNIDX - put invoked function index in Q.
ENA - ENTNUM   - put entry number in A.
RTJ - (ENTPRG) - invoke the new function.
```

The invoked functions are: MATER, RSQ, TRC, MTF own code and the MTF label.

4. OUTPUT PARAMETERS

    a. The output tape (see PSI Excepts No. 69 (Aug. 1974) "Cyberdata Magnetic Tape Subsystem").

    b. The EOD parameters (see EOD table in Cyberdata Software User's Guide).

    c. TAPE I/O ERROR message.

    d. END OF TAPE message.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. ATT EQUATES

        FTOB (1)   - first track of current batch.
        INBUF (5)   - input buffer address.
        TRB (6)   - record buffer.
        RELBA (8)   - relative input buffer address of the current input record.
        CURDA (9)   - current input block disk address.
        OUTBUF (28)   - output buffer address.
        RECORD (31)   - input records counter.
        SVC (36)   - start of SV command area.
        BCOUNT (41)   - mag tape block counter.
        MTCA (46)   - mag tape command area.

    b. MTCA EQUATES

        MTFENT (MTCA)
        MTFEXT (MTCA+1)
        EODADR (MTCA+2)
        MTF (MTCA+3)
        STRBAX (MTCA+8)
        ENDBAX (MTCA+9)
        STATUS (MTCA+10)
        TRANUM (MTCA+11)

        NOTE: see MTCA table in Cyberdata Software User's Guide.

    c. EOD EQUATES

        TOTBLK (2)
        MTFN (3)
        FN (4)
        JN (5)
        BN (8)
        FBLK (9)
        RN (10)
        LBC (12)

        NOTE: see EOD table in Cyberdata Software User's Guide.

d. GENERAL EQUATES

    EODLNG (40)   - EOD record length.
    FMTLNG (10)   - format numbers table length.
    MATER ($0805)   - ENTPRG Q parameter for MATER.
    RSQ ($0806)   - ENTPRG Q parameter for RSQ.
    TRC ($380C)   - ENTPRG Q parameter for TRC.
    MTFOCD ($4800) - ENTPRG Q parameter for own codes.
    LBL ($4820)   - ENTPRG Q parameter for labels.

2. MACROS

    Not used.

3. PROGRAM DEPENDENCIES

    a. Entry points (for ENTPRG):

        ENTRY1 - get next record.
        ENTRY2 - put current record.
        ENTRY3 - write the output buffer.
        ENTRY4 - terminate the function.

    b. Externals

        SETMOV
        PRTMSG
        BD
        MOVREC
        CODPRG

    c. Routines called via transfer vector

        ENTPRG
        EXDFUN
        SUPRW
        TAPIO
        READN
        CLCDWA
        RELFOR

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME __MTFMON - Mag Tape Format Monitor__ __Level 06__

PHYSICAL CHARACTERISTICS

1. PROGRAM SIZE

   732 words.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___OCDMY - Own Code Dummy_____Level 6

PROGRAM FUNCTION

This disk resident function is a dummy own code which is called when an own
code does not exist on disk. OCDMY jumps to ERROR (in GSE) with Q equal error
code $15 (own code is not on disk).

INTERFACE

1. ENTRY METHOD

    OCDMY is called by ENTPRG when a user own code is not found on disk.

2. INPUT PARAMETERS

    None.

3. EXIT METHOD

    OCDMY jumps to ERROR (in GSE) with no return.

4. OUTPUT PARAMETERS OR CONDITIONS

    Error message "OWN CODE IS NOT ON DISK" is displayed.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    ERROR ($96)

2. PROGRAM DEPENDENCIES

    GSE (ERROR). Display error message.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    7 words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ____ OFRZ

PROGRAM NAME ____ Operator Freeze Routine

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   Entry point OFRZ:

   The program puts the batch connected to the station on which the command
   was issued to the "frozen" state. Certain flags are left in the ATT
   image on the first track of batch, the batch status is modified to
   "frozen" in the ABT and the station reverts to the idle state.

2. DETAILED DESCRIPTION

   a. The command is checked for validity.

      If not valid the program exits to ERROR with the appropriate
      error code.

   b. The interrogate and search buffers are released. Necessary disk
      addresses will be saved in these words.

   c. The monitor override check bit is set before the ATT is modified.

   d. If a batch is in End of Data state, and the situation can be
      recognized only be equality of previous disk address in TRB and
      the current disk address in ATT, a flag should be set. Bit 15 in
      the interrogate buffer backwards pointer is set.

   e. Current disk addresses are saved in interrogate and search buffers
      backwards pointers.

   e1. The "format change bit" in TRB header is saved in AINT.

   f. Data which may be lost is written to disk.

      (i)   If the rewrite bit is on, the current record is rewritten. In
            case the record is changed again the rewrite bit is set again.
            Records (not in ENTRY or INSERT mode) can be changed if a field
            is replaced (PSEUDO ENTRY). Such a state may be detected
            also by the revalidation bit.

      (ii) If mode is ENTRY or INSERT and the current record is not
            completed it is written to disk. If record is empty there
            is no need to write it unless it is the first record in ENTRY
            mode, because then disk addresses may be lost.

PAGE NO. 496

AA5641

PRINTED IN THE USA

DOCUMENT CLASS ___OFRZ_____

PROGRAM NAME_____Operator Freeze Routine_____

g. If the batch is in entry mode and under document control, the displacement to the beginning of document is saved in the ATT in order to avoid mistakes in case the LDD is squeezed while the batch is frozen.

The document entry is read from the LDD and the difference between the displacement of the current group to the beginning of LDD and the displacement of the first group is calculated.

h. Accumulated time is updated.

Two words, start time of current run and the accumulated time, are read from first track of the batch. The difference between the accumulated clock and the start time of current run is added to the accumulated time which is then written back to disk.

i. The TRB is released.

j. A flag is set (field count is assigned with $FF) in order to save the End of Field state.

If the format is replaced during the time the batch is frozen, it is not always possible to restore such a state.

k. The Format is released.

The format backwards pointer is cleared in order to indicate to the program which reactivates the batch that no error lock was on the station.

l. The ATT is written in first track of the batch.

The record count is saved as a negative number in order that an ATT modified by the FRZ programs can be recognized.

m. The ABT entry of the batch is updated with the mode "frozen".

n. The station is assigned the IDLE mode.

The screen is cleared and the message BATCH FROZEN appears.

The message is moved from the program to the interrogate buffer and displayed.

o. The switches, basic mode and format number are cleared and the program exits to the dispatcher via CLRSR.

Internal subroutine:

SUBR routine is used to read two words from disk, or write two words on disk.

When entered, A register = sector address.
             Q register = word in sector.

On exit, two temporaries in ATT (word 74, 75) hold the two words which were read from disk. If Q < 0, the temporaries are written to disk.

DOCUMENT CLASS ___OFRZ_____

PROGRAM NAME_____Operator Freeze Routine_____

SUBR is used in order to read:

1. The document entry from LDD.
2. Start time and accumulated time from the first track of batch.

SUBR is used in order to write the updated accumulated time back to disk.

INTERFACE

1. ENTRY METHOD

   Operator enters:

   INT FRZ REL

   The INTRGT program checks that the command was issued on a non-idle station,
   and is responsible for loading the OFRZ program to memory and executing it.

```
   CLR A            first entry point
   LDQ = N$A009     9th interrogate function, load only
   RTJ - (ENTPRG)   load via ENTPRG
   SQN   OK
   JMP   NOCORE
OK ADQ   5,Q        first entry point address
   JMP   1,Q        execute
```

2. EXIT METHOD

   (i) Exit to ERROR if command was not entered correctly with appropriate
        error code.

   (ii) Exit to CLRSR if program finished normally.

3. OUTPUT PARAMETERS OR CONDITIONS

   (i) In case an error is detected in the command format: E1 (invalid command
        format) is displayed.

   (ii) If program finished successfully the message "BATCH FROZEN" is displayed.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. System Status Table

      SST ($47)
      STEMP (20)
      CFFA (12)
      IBL (14)
      TIQ (15)
      CLOCKS (17)

318

DOCUMENT CLASS ___OFRZ_____

PROGRAM NAME_____Operator Freeze Routine_____

b. External System Table

LDDDA (9)

c. ATT

FTOB (1)
CMODE (2)
JIDX (3)
FCNT (3)
RCNT (4)
TRB (6)
PFBA (7)
CURDA (9)
NDA (11)
DT (13)
BMODE (15), BMOD (15)
ELDT (16)
DOCNO (17)
SW1 (19)
SW2 (20)
FUNIDX (21)
ABTIDX (22)
AINT (27)
ASRCH (28)
RCD (32)
ATEMP (73)

d. Low Core Masks

LPMASK (2)
NZERO ($12)
ZERO ($22)
ONEBIT ($23)
ZROBIT ($33)

e. Error Codes

CE1 ($12)

2. MACROS USED

None.

3. PROGRAM DEPENDENCIES

Entry Points:

OFRZ   - Freeze a batch connected to the station on which the command
                  was issued.

EXTERNAL LIST:

DLINE   - Entry in DISPLY
                  Used in order to display informative message BATCH FROZEN.
SETDIS   - Entry in DISPLY
                  Sets display mode to idle.
REGEN   - Entry in DISPLY
                  Used in order to regenerate the display of idle station,
                  i.e. to clear the screen.
GETBUF   - Entry in PIO.
                  Asks to interrogate buffer in order to display a message from it.
MOVREC   - Entry in PIO.
                  Moves the message BATCH FROZEN into the interrogate buffer.
FRZATT   - Entry in PIO.
                  Writes the ATT in the first track of batch.
SETMOV   - Entry in ENEXSR.
                  Enables moving the program during execution.

Routines used by Transfer Vector.

GETALP ($9B)   - Entry in SCANV.
                  Checks that EOR follows the FRZ command.
RELFOR ($AD)   - Entry in GFCORE.
                  Releases interrogate, search, TRB buffers and the format.
REWRIT ($A4)   - Entry in PIO.
                  Rewrites the current record in the rewrite bit in switch 1 is on.
WRITEC ($A3)   - Entry in PIO.
                  Writes current record if the record is not empty, or if it is
                  the first record, and the record is still not written on disk.
SPILL ($A1)   - Entry in SPILL.
                  Changes batch status to frozen in the ABT.
DISP ($A0)   - Entry in DISPLY.
                  Used in order to generate the cleared screen after freezing
                  the batch and to display the message BATCH FROZEN.
ENTSR ($91)   - Entry in ENEXSR.
                  Saves return address in enter/exit stack for the internal
                  routine SUBR.
EXTSR ($92)   - Entry in ENEXSR.
                  Returns via the enter/exit stack from SUBR routine.

DOCUMENT CLASS ___OFRZ___

PROGRAM NAME___ Operator Freeze Routine ___

<pre>
CLRSR ($93) - Entry in ENEXSR.
               Used for exit in case of successful completion of the program.
ERROR ($96) - Entry in GSE.
               Used for exit in case of error in the command format.
CLCDWA ($90) - Entry in PIO.
               Calculates disk word address of document in the LDD and the
               start time and accumulated time in the first track of batch.
REDDWA ($A9) - Entry in PIO.
               Used in order to read from disk the document entry in LDD and
               the start time and accumulated time in the first track of
               batch.
</pre>

4. DATA AREAS AND SYSTEM TABLES REFERENCED

SST
ATT
ABT - active batches tables
LDD - legal documents directory
DTL - (ATT image, start and accumulated time)
Interrogate buffer
TRB

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

Program size - 230 words.
Interrogate buffer - maximum length 42 words.

2. RELOCATION TYPE

Disk Resident.
Run anywhere.

3. ENTRANCY CLASSIFICATION

Re-entrant
Unlocked

321

DOCUMENT CLASS _____ OTHW

PROGRAM NAME _____ Operator Thaw Routine

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    One entry point OTHW.
    The program reactivates a batch which is in "frozen" state. The batch
    status is modified in the ABT. The batch is connected to the station
    on which the command was issued. The ATT of this station is modified
    according to the ATT image in the first track of batch. The ATT image
    can be created by the FRZ routines (supervisor freeze of operator freeze)
    or by the CLR routine (extended power fail recovery routine) which also
    may put a batch to the frozen state. The OTHW program is able to recognize
    an ATT which was written by FRZ according to the negative record count and
    use the ATT image accordingly.

2. DETAILED DESCRIPTION

    a. The station is checked to be idle.

    b. The command parameters: job name, batch number and operator number
        are checked for validity.

    b1. The job is looked for in the LJD. If it is not found, the command is
        rejected. If the job appears in the LJD and the station is a CRT
        the job name is transferred to output queue.

        The following operations are performed under a lock which prevents
        simultaneous modification of the ABT.

    c. The ABT entry specified by the job and batch is looked up. The mode
        in the ABT entry is checked to be frozen. The logical unit in the
        ABT entry is checked to be selected.

    d. The operator number in first track of batch is compared to the operator
        number specified in the command. If not equal the command is rejected.

    e. TRB buffer is allocated.
        If memory is full, command is rejected.

    f. Batch status in the ABT is modified to frozen. The lock is cleared.

    g. The start time of run in first track of batch is updated with the
        current value of the clock.

    h. The station number in the ATT image in first track of batch (word 0)
        is updated with the current station number.

    h. The TIQ (Station input queue) is cleared.

AA5641

DOCUMENT CLASS ___OTHW_____

PROGRAM NAME_____Operator Thaw Routine_____

2. INPUT PARAMETERS OR CONDITIONS

If the station is not idle the command is rejected.

jjjjj   is a job name, 1-6 characters
bbb   is a batch number, 1-999
000   is operator number, 1-999

The job name and batch number should specify an entry of a frozen batch whose
logical unit is selected.
The operator number parameter should be equal to the operator number in first
track of batch.
If the parameters do not fulfill the conditions listed above, the command is
rejected.

3. EXIT METHOD

(i)   Exit to ERROR with appropriate error codes.
(ii)   Exit to CLRSR
       Exit to EORTST
       Exit to VMM1

4. OUTPUT PARAMETERS OR CONDITIONS

(i)   Error codes.
     D1   invalid operator number.
     D6   operator number parameter not between 1 and 999 or is not equal
          to operator number in first track of batch.
     D4   illegal job name.
          Job name parameter is not 1-6 characters or is not found in the LJD.
     D5   illegal batch number.
          Batch number parameter is not between 1 to 999, or the batch
          specified by the job name and batch number parameters is not frozen.
     E1   illegal request.
          The station is non-idle.
     F4   memory full.
          No core is available for the TRB.
     F6   disk deselected.
          The logical unit of the batch is of a deselected disk.

(ii)   Exit to CLRSR if next entered key or character is necessary to specify
       the operation to follow.
       Exit to EORTST if no verification options are specified.
       Exit to VMM1 if in beginning of record.

305

PRINTED IN THE USA

DOCUMENT CLASS ___OTHW___

PROGRAM NAME_____Operator Thaw Routine_____

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. SST

      SST ($47)
      EXTSST (0)
      SLOCK (1)
      BATT (10)
      CATT (11)
      CFFA (12)
      IBL (14)
      TIQ (15)
      CLOCKS (17)
      ATTLNG (19)
      STEMP (20)

   b. EST

      LDDDA (9)
      DSKST (29)

   c. ATT

      CMODE (2)
      JIDX (3), FCNT (3)
      RCNT (4)
      ASCNT (5)
      TRB (6)
      PFBA (7)
      CURDA (9)
      FMTNO (13)
      RFA (14)
      BMODE (15)
      ELDT (16)
      DOCNO (17)
      SW1 (19)
      SW2 (20)
      FUNIDX (21)
      AINT (27)
      ASRCH (28)
      ATOPT (33)
      LUN (35)
      TCLK (36) - Temporary. The current clock is written from this word to start
             time of current run in first track of batch.
      ATEMP (73)

PRINTED IN THE USA

DOCUMENT CLASS ___OTHW_____

PROGRAM NAME_____Operator Thaw Routine_____

Communication Area

COUNTS (36+10)

TJOB (COUNTS)	- 3 temporaries for job name parameter saving.
TBATCH (COUNTS+3)	- 2 temporaries for batch number parameter saving.
TOPR (COUNTS+4)	- temporary for operator number parameter saving.
TEMP (COUNTS+5)	- temporary.
ABTINF (COUNTS+6)	- temporary for first word in ABT saving.

d. Error Codes

CD4	(16)	Illegal job name
CE1	(18)	Illegal request
CD5	(17)	Illegal batch number
CD1	(13)	Invalid operator number
CF6	($1E)	Disk deselected
CFU	($17)	Memory Full

2. MACROS USED

None.

3. PROGRAM DEPENDENCIES

Entry points

OTHW - reactivates a batch on the station on which the command was issued.

External List

REGEN	- Entry in DISPLY.
	Used in order to regenerate the screen with the last processed record of the reactivated batch.
SETDIS	- Entry in DISPLY
	Sets display mode to DATA mode.
STTYPE	- Entry in EMM.
	Sets data type in ATT according to the format.
GETLJD	- Entry in GETLJD.
	Gets the JOB entry which is specified in the command.
SETMOV	- Entry in ENEXSR.
	Used in order to enable the program to move.
BD	- Entry in ARITH.
	Converts the binary value of the operator number to packed decimal.
EORTST	- Entry in VMM.
	Program exits to End Of Record Test if no verification should be done on this field.
VMM1	- Entry in VMM.
	Program exits to check field entry if it is beginning of record and no special key has to be entered.

3c 7

DOCUMENT CLASS ___OTHW_____

PROGRAM NAME_____Operator Thaw Routine_____

Routines used by Transfer Vector.

CETALP ($9B) - Entry in SCANV.
                Retrieves and analyzes the job name parameter.

GETBIN ($99) - Entry in SCANV.
                Retrieves and analyzes the batch and operator number.

ENTSR ($91) - Entry in ENEXSR.
                Saves return address for the internal I/O routine Subr.

EXTSR ($92) - Entry in ENEXSR.
                Gets the return address for the internal I/O routine SUBR.

CLRSR ($93) - Entry in ENEXSR.
                Used in order to exit to dispatcher when the batch is waiting
                for next character or key to come.

WAIT ($B0) - Entry in ENEXSR.
                Delays for one cycle the execution of the part which checks
                and modify the batch status if the lock bit in the SST is
                already set.

CLCDWA ($90) - Entry in PIO.
                Calculates the word disk addresses of:
                1.   Operator's number in first track of batch.
                2.   Document's entry in LDD.
                3.   Start time of current run in first track of batch.
                4.   Station number in ATT image in first track of batch.
                5.   Several parts in ATT image in first track of batch
                   (when reading them to core).

REDDWA ($A9) - Entry in PIO.
                Reads from disk:
                1.   Operator's number in first track of batch.
                2.   Document entry in LDD.
                3.   Several part in ATT image in first track of batch.

WRTDWA ($AA) - Entry in PIO.
                Writes to disk:
                1.   Start time of current run.
                2.   Station number in the ATT image.

GFCORE ($9D) - Entry in GFCORE.
                Gets into memory the necessary format for the reactivated batch.

RELFOR ($AD) - Entry in GFCORE.
                Releases the interrogate buffer.

CAM ($AE) - Entry in CAM.
                Allocates memory for the TRB.

READN ($A6) - Entry in PIO.
                Reads into the TRB the last processed record.

ERROR ($96) - Entry in GSE.
                Used for exit in case of failure.

SPILL ($A1) - Entry in SPILL.
                Used in order to get and modify the ABT entry.

DOCUMENT CLASS ___OTHW_____

PROGRAM NAME_____Operator Thaw Routine_____

4. DATA AREAS AND SYSTEM TABLES USED

    SST
    ATT
    TRB
    LDD
    DTL - ATT image, start time of current run, operator number.
    TOPT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    Program size - 490 words
    Interrogate buffer - 45 words
    TRB - length as specified in the job entry minimum 9 words.
    Format - length of format whose format number appears in the ATT minimum
          7 words.

2. RELOCATION TYPE

    Disk resident.
    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Re-entrant.
    Partially locked.

309

DOCUMENT CLASS ___PBR___

PROGRAM NAME ___Print Batch Report___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   PBR processes the supervisor request to print batch report from the
   statistics file according to the following parameters:

   a. PBR   Include all batches in the statistics file

   b. PBR,J,XXXXXX   Include all batches for job XXXXXX

   c. PBR,O,XXX   Include all batches entered by operator XXX

2. DETAILED DESCRIPTION

   PBR is an overlay called in by ENTPRG, to process the supervisor
   function PBR. First PBR validates the command parameters. If a
   parameter error is found, an invalid parameter of invalid job name
   error message is generated via SUPER (REJSUP). If a command format
   error is found, an invalid command error message is generated. If
   no errors are found, PBR sets ATT/TEMP+2 to FFFF for all batches

                              1   for job
                              0   for operator,

   PBR makes a request of CAM for a sector size buffer. It calls PIO
   (GETBUF) to allocate, an interrogate buffer of 48 words. The first
   sector of the statistics file is read into core. The number of records/
   file is initialized. Each record is examined to see if it should be
   included in the report (all, job name, or operator number). If the
   information should be included, a first time check is made to see if
   the header should be output first. If the list device is a 970-480,
   a check is made to see if the screen was full. If so, CONCAN is
   called to see if the operator wishes to continue with the report or
   cancel the request. If cancel, (TOEXIT) the sector size buffer is
   released and exit is via REJSUP entry pointer in the supervisor command
   processor SUPER. If continue, the cursor is reset via PIO (SUPRW) and
   the header is displayed.

   (ALL) All the data included in the batch report (job name, operator
   number, batch number, start time, run time, number of records, key
   strokes per hour, error records, and mode) is extracted from the current
   statistics record, formatted, and placed in the output buffer. Internal
   subroutines BINASC and BCDASC are used for binary to ASCII and BCD to
   ASCII conversations. The buffer is printed via PIO (SUPRW). If operator
   entered SX in attempt to cancel the report, exit at TØEXIT then checks
   are made to see if this is the last record in the statistics file, at
   which time the program exits at PBREXT. If it is the last record in this
   sector, the next statitistics file sector is read in. Processing continues
   for each record as for the first.

Λ

DOCUMENT CLASS ___PBR_____

PROGRAM NAME_____Print Batch Reports_____

INTERFACE

1. ENTRY METHOD

   PBR is a supervisor function processor loaded by ENTPRG. Execution begins at
   its only entry point PBR.

2. INPUT PARAMETERS OR CONDITIONS

   I = current ATT base address
   Operator enters either
       PBR
       PBR,J,XXXXXX
       PBR,O,XXX

3. EXIT METHOD

   Release the sector size buffer
   Jump to (SV) or JMP to REJSUP with (A)239 (function terminated).

4. OUTPUT PARAMETERS OR CONDITIONS

   Batch report printed.
   I = current ATT base address

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE:

       LPMASK (2)
       NZERO ($12)
       ZERO ($22)
       ONEBIT ($23)
       ZROBIT ($33)
       TEN ($46)
       SST ($47)
       ENTSR ($91)
       EXTSR ($92)
       SUPRW ($97)
       GETBIN ($99)
       GETALP ($9B)
       GETCOM ($9C)
       REDSEC ($A7)
       CAM ($AE)
       SV ($B1)
       REJSUP ($B2)

2̶9̶5̶

DOCUMENT CLASS ___PBR_____

PROGRAM NAME_____Print Batch Report_____

    b. SST:

       STEMP (SST+20)

    c. ATT:

       TERMIN (23)
       AINT (27)
       SCA (36)
       TEMP (50)
       JN (58)
       STATUS (61)

    d. SRL:

       SRJOB (3 + 0)
       SRBCH (3 + 1)
       SROPR (3 + 2)
       SRSTRT (3 + 3)
       SRACCT (3 + 5)
       SRSTRK (3 + 6)
       SRERCS (3 + 8)
       SRRECS (3 + 9)
       SRJN (3 + 13)

    e. EST:

       SFDA (5)
       RCSSF (7)

    f. CONSTANTS:

MOT (5)	Motion request code
FW (3)	Format write code
J ($4A)	ASCII J
O ($4F)	ASCII O

3. PROGRAM DEPENDENCIES

    PIO - (SUPRW) - Read from/write to supervisor devices
        - (REDSEC) - Read a disk with sector addressing
        - (GETBUF) - Get a terminal interrogate buffer
    CONCAN - Wait for continue or cancel key by operator
    ARITH - (DB) - Decimal to binary conversion
    GETLJD - Locate a job in the LJD
    SCANV - (GETBIN) - Get binary number from input string
        - (GETALP) - Get alpha character from input string
        - (GETCOM) - Get comma from input string
    CAM - Allocate core buffer
    SUPER - (SV) - Conclude supervisory function request
        - (REJSUP) - Generate error message to supervisor

296

DOCUMENT CLASS ___PBR___

PROGRAM NAME___Print Batch Report___

     ENEXSR - (ENTSR) - Common subroutine entry
           - (EXTSR) - Common subroutine exit

4. DATA AREAS USED

   Interrogate buffer
   ATT
   SRL

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $1A6_{16} = 422_{10}$ words

2. RELOCATION TYPE

   Run anywhere

3. ENTRANCY CLASSIFICATION

   Not reentrant

ATTACH ANALYTICAL SUPPLEMENTS

Sample batch report for all:

JOB	OPR	BCH	STRT	RNTM	RECDS	KPH	ERECS	MODE
TEST	007	001	0931	2	14	49	0	ENTR
TEST	033	055	0941	1	4	24	0	ENTR
TEST	033	056	0943	1	5	43	0	ENTR
TEST	033	055	0941	1	4	43	0	ENTR
TEST	077	100	1330	1	6	62	0	ENTR
TEST	077	200	1334	1	7	84	0	ENTR
SIGN	007	001	1116	1	3	57	0	ENTR
ASSEM	007	001	0853	24	59	24	0	ENTR
ASSEM	007	001	0924	6	59	35	0	VERI
PAT	033	001	1859	41	91	30	0	ENTR
PAT	033	002	1011	1	2	37	0	ENTR
ASSEM	007	003	1038	3	1	7	0	ENTR
PAT	033	001	1941	5	91	6	0	VERI
ASSEM	007	001	1943	5	22	41	0	ENTR
TEST	007	200	2007	1	7	14	0	VERI
TEST	005	200	2011	0	7	9	0	VERI
NAME	007	001	0924	1	6	78	0	ENTR
NAME	016	001	0925	1	6	31	0	VERI
NAME	015	001	0926	1	5	34	0	VERI
PAYROL	001	001	1258	9	19	24	1	ENTR
PAYROL	001	001	1308	4	19	22	1	VERI
PAYROL	001	001	1315	1	19	15	0	VERI

.297

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___PBS - Print Batch Status_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    This program performs the supervisor command: PBS - Print Batch Status.

    Command syntax is:

                    PBS,aaaaaa(1) CR

    or: PBS CR

                    where (1) - job name

    This command is used to print the batch number, mode, verification
    options, and error status of all batches in a specified job (when
    job name is given), or of all batches in the system (when job name
    is omitted). It also prints the number of the disk the batch resides
    on.

2. DETAILED DESCRIPTION

    The program checks if a job name was given in the command. If it was
    not, PBS2 is called via ENTPRG, and control is passed to it.

    If a job name appears, it is checked for legality. GETLJD is called
    and returns with the job index in A register. If the job does not
    exist (Q -ve) exit is made to REJSUP. Spill is called and returns the
    entry in ABT for the job's batches, starting with batch 0.

    Before batch status is printed, CRT status is checked. If end of
    screen was reached CONCAN is called to allow the operator to cancel
    the request. If a new screen is started then the header: "Job
    Batch Mode Ver-Op Status Disk" is displayed, under which the details
    of the batch are displayed. If it is not yet end of screen, only
    the details of the batch are displayed. The program checks if a
    cancel was received, and if so transfers control to REJSUP.

    Any line to be printed is moved to the interrogate buffer (where
    blanks were stored before), and printing is performed by SUPRW
    (entry is PIO). In any case of error REJSUP is called. When all
    the batches in the job have been processed (SPILL returns with Q
    negative), control is returned to SV.

INTERFACE

1. INPUT PARAMETERS OR CONDITIONS

   I = current ATT base address
   Job name (if any), in Interrogate buffer.

2. EXIT METHOD

   Return to SV   - when no errors detected and no cancel was received
                    during execution.

   Exit to REJSUP - in case of error or cancel.

3. OUTPUT PARAMETERS OR CONDITIONS

   I = current ATT base address

   Error messages:

   a. Invalid job name CAPS

       The job name contained more than 6 characters or a comma, or a
       non-ASCII character appeared in the job name.

   b. Undefined job name - CAPS

       The job name does not exist in LJD.

   c. Invalid command format - CAPS

       A non-ASCII character appeared in command, or a character other
       than a comma or a blank or EOR after the command name.

   d. Invalid parameter - CAPS

       A non-ASCII character in the job name or a comma in the job name.

   e. Function terminated - CAPS

       The CANCEL key was entered, or MI SX was entered from the
       Teletype.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ONEBIT ($23)
        GETALP ($9B)
        REJSUP ($B2)
        GETCOM ($9C)
        SUPRW ($97)
        SPILL ($A1)
        SV ($B1)
        ENTPRG ($9E)
        EXDFUN ($9F)
        ENTSR ($91)
        EXTSR ($92)
        STEMP ($5B)

    b. ATT

        TEMP (50)
        AINT (27)
        SCA (36)
        CANC (23)
        CRTSTS (62)

    c. CONSTANTS

        MOT (5)
        FW (3)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

    PIO   -
    ENTSR - Store return address in stack
    EXTSR - Return via stack
    SPILL - Retrieve ABT entry
    SV   - Normal return
    ENTPRG - Call PB2 if no job name specified
    CONCAN - Wait for CANCEL or CONTINUE
    SETMOV - Enable program move
    GETLJD - Check for valid job name.

DOCUMENT CLASS   CYBERDATA IMS

PROGRAM NAME   PBS - Print Buffer Status   Level 6

4.   DATA AREAS USED

     ATT
     SST
     ABT
     INT. BUFFER
     LJD

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___PB2___

PROGRAM NAME___Print Batch Status - Part 2___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   PB2 processes the PBS CR statement by scanning the Active Batch Tables, and
   writing a status line for each active entry.

2. DETAILED DESCRIPTION

   a. (PB2). Call CAM to reserve 99 words of core.

      ● If core is not available, exit to REJSUP with (A) = 11 (MEMORY FULL -
        RETRY).

      ● Set up linkage to reserved area.

      ● Set ATT/CRTSTS = $7FFF.

      ● Set ATT/TEMP + 6 = 0 (Job Index).

      ● Set ATT/TEMP + 10 = 0 (ABT cumulative sector index).

   b. (NXTABT). Set ATT/TEMP + 1 = 0 (ABT sector index). Set ATT/TEMP + 2 = 3
      (ABT entry index).

   c. (NXTSEC). Call REDSEC to read SPIL sector addressed by ATT/TEMP + 1 plus
      ATT/TEMP + 10.

      ● If ATT/TEMP + 1 = 0, save batch track address in ATT/TEMP + 3.

   d. (NXTENT). If current ABT entry is inactive, go to Step 11.

      ● Call CHCSTS to write header to output unit, if necessary, and to blank-fill
        Interrogate buffer.

      ● If current batch is part of same job as preceding batch, go to Step e.

      ● If not same job, add the difference between current Job Index and preceding
        Job Index to ATT/TEMP + 6.

      ● Call GETLJD to return information for this job.

      ● If the job does not exist, clear user count in core buffer, and exit to
        REJSUP with (A) = 5 (UNDEFINED JOB NAME).

      ● Save job name in ATT/TEMP + 7 through ATT/TEMP + 9.

   e. (SAMJOB). Move job name to Interrogate Buffer positions 0-5.

      ● Convert Batch Number to ASCII representation of hexadecimal number and
        store it in Interrogate Buffer positions 7-9.

369

DOCUMENT CLASS __PB2__

PROGRAM NAME___Print Batch Status - Part 2___

- ● Store a 4-character mnemonic in Interrogate Buffer positions 12-15 to represent Batch mode:

  ENTR   Entry mode
  WAIT   Entry completed, waiting for verification
  EFRZ   Entry frozen
  VERI   Verify mode
  VFRZ   Verify frozen
  COMP   Completed
  WRIT   Written
  MAIN   Maintenance

- ● Store a series of 1-character codes in Interrogate Buffer to indicate Verification type:

  F   Verify according to format (position 18)
  E   Verify erroneous fields (position 20)
  U   Verify fields with unbalanced registers (position 22)

- ● Store a series of mnemonic codes in Interrogate Buffer to indicate Batch status:

  U   Batch unbalanced (position 25)
  B   Batch balanced (position 26)
  ER   Error-flagged record in batch (positions 28-29)

- ● Call SUPRW to write the line formatted in the Interrogate Buffer, and continue with Step k.

f. (CHCSTS). Call ENTSR to save return address in Exits stack.

- ● If ATT/CRTSTS = $7FFF (first time), go to Step i.

- ● If output unit is terminal, go to Step g.

- ● If Terminate key was entered, clear user count in core buffer and exit to REJSUP with (A) = 39 (FUNCTION TERMINATED message).

g. (ON). If ATT/CRTSTS ≠ $C000 (Error condition) go to Step j.

h. (NXTSTR). Call CONCAN to get next input character.

- ● If CANCEL key was entered, exit to REJSUP with (A) = 33.

- ● If next input character is not CONT, repeat Step h.

i. (MOTION). Call SUPRW to write EOF and the header
JOB BATCH MODE VER-OP STATUS
on the output unit.

DOCUMENT CLASS ____PB2____

PROGRAM NAME____Print Batch Status - Part 2____

    j. (BLNKS). Call BBBBBB to blank-fill the Interrogate Buffer (15 words).

      ● Exit to caller via EXTSR.

    k. (ADVANC). If ATT/TEMP + 2 = 93 (End of ABT sector), go to Step 1.

      ● Increment ATT/TEMP + 2 by 3 (Advance to next ABT entry), and continue at Step d.

    l. (LSTENT). Increment ATT/TEMP + 1 by 1 (Advance to next ABT sector).

      ● If not at end of ABT, set ATT/TEMP + 2 = 0, and continue at Step c.

      ● If not last ABT, go to Step n.

    m. (PB2EXT). Clear user count in core buffer, and exit to SV.

    n. (NOLAST). Set ATT/TEMP + 10 = ATT/TEMP + 1 plus ATT/TEMP + 10 (increment cumulative sector index) and continue at Step b.

INTERFACE

1. ENTRY METHOD

PB2 is a disk-resident program which is called via ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

(I) = Base address of Active Terminal Table (ATT).

3. EXIT METHOD

Normal exit is to SV.
Abnormal exit is to REJSUP.

4. OUTPUT PARAMETERS OR CONDITIONS

(I) = Base address of ATT.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

      LPMASK ($2)
      ZERO ($22)
      ONEBIT ($23)
      SST ($47)
      ENTSR ($91)
      EXTSR ($92)
      SUPRW ($97)

371

PRINTED IN THE USA

        GETSTR ($A2)
        REDSEC ($A7)
        CAM ($AE)
        SV ($B1)
        REJSUP ($B2)

    b. SST

        STEMP (SST + 20)

    c. EST

        SPIL (23)

    d. ATT

        JIDX (3)
        CANC (23)
        AINT (27)
        SCA (36)
        TEMP (50)
        CRTSTS (61)

    e. SUPRW Codes

        FW (3)
        MOT (5)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

ENEXSR	ENTSR	Save return address in Exits stack
	EXTSR	Exit to last address in Exits stack
	SUPRW	Supervisor Read-Write
PIO	GETSTR	Get a string of characters from input buffer
	REDSEC	Read sector
CAM	CAM	Core allocation module
SUPER	SV	Supervisor command processor
	REJSUP	Reject supervisor command
GETLJD	GETLJD	Locate an entry in Legal Job Directory
CONCAN	CONCAN	Wait for Continue or Cancel Key.

4. DATA AREAS USED

SST	System Status Table
EST	Extended System Table
ATT	Active Terminal Table
ABT	Active Batch Table

DOCUMENT CLASS ___PB2_____

PROGRAM NAME_____Print Batch Status - Part 2_____

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $12B_{16} = 299_{10}$ words.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

373

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____PDC - Print Documents_____Level 6

PROGRAM DESCRIPTION

1. GENERAL DESCRIPTION

    PDC is a supervisor function called under supervisory mode in order to
    display documents information.

    Calling sequence is PDC   to display a list of the documents in the system.

                         PDC,aaa   to display the formats associated with document
                              aaa   followed by number of records per format

2. DETAILED DESCRIPTION

    PDC first enters SETMOV to enable program moving and goes to extract the
    fourth character entered by the supervisor. If it is not end of record or
    comma or a space goes to REJSUP to indicate "Invalid command format"
    (ERR9). If it is space looks for end of record via GETCOM and goes to
    ERR9 if not found. When end of record is found goes to DOCLST to display
    list of the documents in the system.

    When a comma is found, enters GETBIN to get the document number, and
    checks it against the largest document number possible (255). If it
    is illegal goes to REJSUP to indicate "Illegal document number" (ERR21).
    If it is a legal document number goes to GETENT subroutine to read the
    requested document entry in the LDD. If the first word of the entry is
    zero, goes to REJSUP to indicate inactive document. If the entry is not
    zero, issues core request to CAM according to the length specified in the
    LDD entry, enters TOPSR subroutine in order to set the display device to
    start point, requests interrogate buffer for the output string and fills
    it with blanks through BLANKS subroutine. Then prepares parameters and
    enters GETENT subroutine to read the requested document from the LDT,
    sets pointers for the document buffer and display buffer and starts a
    loop to extract groups of format number and number of records, converts
    them to ASCII via BINASC subroutine and stores them to the output buffer.
    When the output buffer is filled, enters PRINT subroutine to display the
    current line, and handle SX or cancel requests and end of screen status.
    When the last group of format number/number of records is found (bit 15
    of the format number is set) goes to display the last line and exits to
    SV.

DOCLST. When a list of the documents in the system is requested, issues a core request of 99 words (sector size + header) and interrogate buffer and fills it with blanks through BLANKS subroutine. Prepares parameters and reads sector of the LDD via REDSEC. Starts a loop to check the first word of a document entry in the LDD. When it is not zero and it is the first time sets the display device to start point. Moves the header to the output buffer and displays it. Converts the document sequence number to ASCII via BINASC subroutine and stores it in the output buffer. When the output buffer is filled, displays it through PRINT subroutine and then resets the output buffer pointer. When the document sequence number is 256, displays the last line if required, releases the requested core, and exits via SV. Else when end of LDD sector is detected goes to read the next sector (NXTSEC) and if not at end of sector goes to handle the next entry (NXTDOC).

INTERFACE

1. ENTRY METHOD

    PDC is a supervisor function loaded by ENTPRG.
    Only entry point is PDC

2. INPUT PARAMETERS OR CONDITIONS

    I = ATT address
    Operator enters either -

    PDC
    PDC,aaa

3. EXIT METHOD

    1. Jump to REJSUP with appropriate error codes.

    2. Jump to SV.

4. OUTPUT PARAMETERS OR CONDITIONS

    I = ATT address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK ($2)
      ZERO ($22)
      ONEBIT ($23)
      TEN ($46)
      SST ($47)

   b. SST

      STEMP (20)

   c. TRANSFER VECTOR

      CLCDWA ($90)
      ENTSR ($91)
      EXTSR ($92)
      SUPRW ($97)
      GETBIN ($99)
      GETALP ($9B)
      GETCOM ($9C)
      REDSEC ($A7)
      REDDWA ($A9)
      CAM ($AE)
      SV ($B1)
      REJSUP ($B2)

   d. EST

      LDDDA (9)

   e. ATT

      TERMIN (23)
      AINT (27)
      SCA (36)
      TEMP (46)

   f. DOCUMENT

      FMT (3)
      RECS (4)

   g. LDD

      DOC (3)

   h. I/O REQUEST CODES

      FW (3)
      MOT (5)

2. PROGRAM DEPENDENCIES

   a. EXTERNALS

      GETBUF - Interrogate buffer request subroutine (in PIO)
      CONCAN - Wait for continue or cancel key
      SETMOV - Set indications to enable program moving

   b. PIO   - CLCDWA - Calculate disk word address
            SUPRW - Perform I/O from/to Supervisor devices
            REDSEC - Read a sector from disk
            REDDWA - Read n words from disk according to word address

   c. ENEXSR - ENTSR - Common subroutine entry
             EXTSR - Common subroutine exit

   d. SCANN - GETBIN - Get binary number from input string
            GETALP - Get alpha character from input string
            GETCOM - Get next comma or end of record from input string

   e. CAM   -   Allocate core buffer

   f. SUPER - SV   - Conclude supervisory function request
            REJSUP - Generate error message to supervisor

3. DATA AREAS AND SYSTEM TABLE USED

   SST
   EST
   ATT

   Interrogate buffer (requested via GETBUF) for output.
   Document or LDD input buffer (requested via CAM).

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $16D_{16} = 365_{10}$ for program.

   99_{10} words buffer for Document list.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME___PER - Print Erroneous Records_____Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    PER is a supervisor function called under supervisory mode to print
    erroneous records in completed batches, or to print all records of
    a batch marked as erroneous in its ABT entry.

    Calling sequence is - PER,JJJJJJ,aaa,bbb,(A)

        JJJJJJ is job name, aaa is starting batch, bbb is ending batch and
        the optional A indicates to print all the records in each erroneous
        batch.

2. DETAILED DESCRIPTION

    PER first enters SETMOV to enable program moving, then collects the
    parameters entered via GETALP and GETCOM in SCANV, checks that commas
    separate between the parameters, and that characters of the job name
    are per alpha; else goes to REJSUP to indicate "invalid command format"
    (ERR9). The batch numbers are checked to be legal numbers not exceeding
    1000, else goes to REJSUP to indicate "invalid batch number" (ERR4). When
    the optional parameter is used, it is checked to be an A otherwise
    indicates "illegal parameter" via REJSUP (ERR3). The parameter string
    must conclude with end of record or blanks and end of record, else goes
    to ERR3. The A (all) option is marked by a flag (ATT/PLLFLG); then checks
    in the PHYSTB that the supervisory list device is line printer or teletype;
    another device is rejected by going to REJSUP indicating that "command
    invalid on this station". Then decides the length of the output buffer
    according to the list device, and requests an interrogate buffer of that
    length via GETBUF. Prepares parameters and enters GETLJD to extract the
    entry of the requested job from the LJD. If there is no entry for that
    job, goes to REJSUP to indicate "job does not exist". Requests core for
    TRB if the length specified as "record length" in the LJD entry, and
    constructs a request to SPILL to find the batch entry in theABT (GETABT).
    If the batch is not found or it was the last batch in that job (A≤0; Q<0)
    goes to PEREXT where the current format is released if a format was called,
    the TRB area is released and if a message is required goes to REJSUP; else,
    goes to SV. If the batch is not found but it is not the last batch in that
    job (A 0; Q 0) goes to ADVANC where checks if it is the last requested
    batch, in which case goes to PEREXT; else, steps to the next batch and
    loops back to GETABT. If the batch is found in the ABT, saves the ABT
    information and checks the batch mode; for modes other than entry-completed,

verify completed and written goes to ADVANC. When it is one of these modes,
constructs and prints a header line for the batch, and if SX (function
cancellation) is requested, goes to PEREXT. Then if there is no erroneous
bit in the batch information in the ABT (WORD1/8) goes to PRTMSG to
indicate "no erroneous records", if the bit is set checks that disk drive
of the batch is selected. If not, goes to REJSUP to indicate that "Disk
XX not selected". If disk is selected, the batch status is changed to
maintenance via SPILL, the first track of batch (ATT/FTOB) is converted to
word addressing and READN is entered to read a record. (IFFRST) If it is
not the first record in the batch, enters READP to read the preceding
record and returns to IFFRST. From the header of the first record extracts
the format number and calls it via GFCORE. If the record is not an
erroneous record and "Print All" option is not specified, goes to NXTREC
where a decision is made whether to print the last line. If it is the
last record in the batch, the batch mode is returned to its original state
via SPILL. If no erroneous records are found in the batch prints "No
erroneous records" and goes to ADVANC. When it is not the last record
reads the next record via READN, if the format is changed goes to GETFT1
to call the new format, else to RELLUP to handle the new record. For
erroneous records prints "Record no. XXXXX" with the sequence number of the
record in the batch and moves the record to the output buffer according
to the fields definition in the format. Erroneous fields are preceded by
"**", and a blank is inserted between fields. Each character is stored
via the STCHR subroutine where according to a flag it is decided to store
it in the most significant or the least significant half of the word, the
output buffer pointer (ATT/WRDCNT) is updated and when it points to the
buffer limit goes to print the buffer and returns.

INTERFACE

1. ENTRY METHOD

    PER is a supervisor program loaded by ENTPRG. Only entry point is -PER.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT address.

    Operator enters either -

        PER,jjj,bbb,bbb
        PER,jjj,bbb,bbb,A

3. EXIT METHOD

    Jump to REJSUP with appropriate error code.
    Jump to SV.

4. I = current ATT address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK ($2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        TEN ($46)
        SST ($47)

    b. SST

        STEMP (SST+20)

    c. TRANSFER VECTOR

        CLCDWA ($90)
        ENTSR ($91)
        EXTSR ($92)
        SUPRW ($97)
        GETBIN ($99)
        GETALP ($9B)
        GETCOM ($9C)
        GFCORE ($9D)
        SPILL ($A1)
        READP ($A5)
        READN ($A6)
        RELFOR ($AD)
        CAM ($AE)
        SV ($31)
        REJSUP ($B2)

    d. EST

        DSKST (29)

e. ATT

    FTOB (1)
    ATRB (6)
    PFBA (7)
    FMTNO (13)
    CANC (23)
    AINT (27)
    ABTIDX (22)
    BATCH (35)
    SCA (36)
    LU (SCA+1)
    ATEMP (73)
    TEMP (46)
    OP (TEMP)
    WRDCNT (TEMP+1)
    FLD (TEMP+2)
    T (TEMP)
    ALLFLG (T+4)
    OPLIM (T+5)
    JOB (T+6)
    ERRFLG (T+7)
    SB (T+8)
    EB (T+9)
    FORMAT (T+10)
    RECNT (T+11)
    INP (T+12)
    CHRCNT (T+13)
    RAOFLG (T+14)
    ABTSTS (T+15)
    EFLAG (T+16

f. CONSTANTS

    A ($41)
    FW (3)
    EREQST (8)

2. PROGRAM DEPENDENCIES

   EXTERNALS

   GETLJD - locate job in LJD.
   GETBUF - allocate interrogate buffer.
   BD   - convert binary number to BCD.
   PRTMSG - print message to the supervisor.
   SETMOV - set indications, to enable program move.
   PIO   - CLCDWA - calculate disk word address.
            SUPRW - perform I/O from/to supervisor devices.
            READP - read a record from disk according to previous record.
                     address.
            READN - read a record from disk according to next record address.
   ENEXSR - ENTSR - common subroutine entry.
            EXTSR - common subroutine exit.
   SCANV - GETBIN - get binary number from input string.
            GETALP - get alpha character from input string.
            GETCOM - get next comma or end of record from input string.
   GFCORE - GFCORE - get format into core.
            RELFOR - release a format.
   SPILL -   - handle Active Batch Table.
   CAM   -   - allocate core buffer.
   SLLPER - SV   - conclude supervisor program.
            REJSUP - generate error message to supervisor.

3. DATA AREAS AND SYSTEM TABLES USED

   SST
   EST
   ATT
   FMT
   TRB
   ABT
   Interrogate buffer (requested via GETBUF)

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___PER - Print Erroneous Records___ Level 06

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $1FO_{16} = 496_{10}$ words.
   Interrogate buffer of 72 or 132 characters.
   TRB according to record length specified in the job.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    To process the PFM command.

    The print format command is given in the following form:

        PFM   ,nnnn(1)   ,nnnn(2)

    where:

        (1)   number of first format
        (2)   number of last format

    If no format numbers are specified all the format numbers in the system,
    together with an indication whether each format is active or inactive,
    are printed. If one format number is specified, its parameters are
    printed. If two format numbers N1, N2 are specified, the parameters of
    formats N1 through N2 are printed. (For detailed description of output
    format and parameters see "Cyberdata System Reference"). PFM is also
    used by PFT for printing format specifications.

2. DETAILED DESCRIPTION

    The validity of the PFM command is checked: in case of syntax error,
    return is made to supervisor mode and a message is printed. A buffer
    of sector size is requested. If core not available, return is made to
    supervisor mode with a message.

    Now the program divides into two: <u>Part 1</u> handles the case in which no
    format number is specified; <u>Part 2</u> handles the case whereby one or two
    format numbers are specified.

    <u>PART 1</u>

    One sector of LFD is read with the buffer (by REDSEC; the base address
    of LFD in entry 39 of EST).

    The I/O bit set before reading is cleared allowing the buffer area to
    move. A header is printed (by use of <u>TRANSF</u> and <u>SUBR</u>).

The LFD is scanned sequentially and for each entry (format) an A or I is printed accordingly to bit 15 in the 2nd word of entry. Next the format number is printed. (Actually this information is inserted in interrogate buffer.) Before information is inserted a check is made if interrogate buffer is full. If it is full it is printed (by use of <u>SUBR</u>) - after each line is printed <u>CRTCHK</u> is invoked to check if end of screen occurred or if cancel was entered (for detailed description of CRTCHK see "Internal Subroutines"). Skip is made over zero entries in LFD (nn - existed format) after each entry has been processed the program checks to see whether the whole sector has now been scanned; if so, a new sector is read into the buffer.

If the last entry in the LFD was processed the remaining information in the interrogate buffer is printed out and the program exits.

<u>PART 2</u>

A syntactic check is made; in case it is a line printed GETBUF is invoked to ensure a 84 characters buffer. (LUFLAG is the indicator of output device). Next the corresponding part of the LFD is read into the buffer (via REDDWA). The corresponding format address being calculated by CLCDWA.

The corresponding entry in LFD is checked; if it is a zero entry (nn - existent format) and only one format number was specified, the program exits with error. If more than one format number was specified, pointers are set up to corrected values and the next entry in LFD is checked. If the entry is non-zero the format is brought into core (by use of GFCORE, format number in Q).

A new screen is moved in (each format begins at a new page) and the header is transferred to the output buffer.

The first line of information is inserted into the output buffer (in the header) printed (by SUBR).

The second header is inserted into the buffer and printed.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME____PFM - Print Format_____Level 06

The name field is obtained via GFNAME (may be empty field). The format
information (data type, number of characters, keying mode, recording
mode, Verify conditions, auto. function, non-zero test, auto. error
flagging) up to the 4th word is inserted in the output buffer by moving
word by word. Since the parameters in the 4th, 5th and 6th words may be
present or not (if the associated bit in the 3rd word is set or not), the
corresponding information is retrieved by use of FMTPAR (the information
returned in A), and is inserted in the output buffer by use of ASCII.
The specification of counters is checked by FMTPAR and if present is
processed by CNTR. If the limits parameter is not present, an "N" is
inserted in the output buffer, otherwise the parameter, whether field
has to be between the limits or outside them.

The printout of the limit parameters differs if the output device is a
line printed or a CRT. If it is a CRT (depends on LUFLAG), the output
buffer is printed and the limits parameters are printed on the next line,
(there is need to check for end of page condition.), preceded by the
header "LIMITS". In case of line printed it appears on the same line as
the other parameters. The limit parameters are inserted in the output
buffer by MOVLIM. The line is printed. It is checked if it was the last
field. If it was the last field of the format after setting up pointers,
CRTCHK is invoked to check for end of screen condition (or cancel) in which
case a new header is printed and processing of a new field begins.

If it was the last field, a check is made to see if it was the last
format requested. If not, processing of a new record begins after checking
for the cancel command.

If it was the last format the program exits.

INTERFACE

1. ENTRY METHOD

   PFM is loaded by ENTPRG. Execution begins at entry point PFM. The second
   entry point, is used by PFT.

2. INPUT PARAMETERS

   Given with the command (see general description).

3. EXIT METHOD

   a. Completion Exit

      JMP - (SV)

   b. Error Exit

      JMP - (REJSUP)

   c. No Core

      ENA 9
      JMP - (REJSUP)

   d. Cancel Entered

   e. If PFM was invoked by PFT, the parameter CALRPI,I will be nonzero
      and at the end of each format, or if cancel was entered, return
      is made to PFT at its second entry point, after releasing PFM via
      EXDFUN.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK ($2)
      ZERO ($22)
      NZERO ($12)
      ONEBIT ($23)
      ZROBIT ($33)
      TEN ($46)
      CLCDWA ($90)
      ENTSR ($91)
      EXTSR ($92)
      SUPRW ($97)
      GETBIN ($99)
      GETCOM ($9C)
      GETALP ($9B)
      GFCORE ($9D)
      ENTPRG ($9E)
      EXDFUN ($9F)
      REDSEC ($A7)
      REDDWA ($A9)

RELFOR ($AD)
CAM ($AE)
SV ($B1)
REJSUP ($B2)

b. ATT, SST, EST

PFBA (7)
FMTNO (13)
RFA (14)
CANC (23)
AINT (27)
SST ($47)
STEMP (SST+20)
CFFA (12)
MAXFMT (24)
LFDDA (39)

c. SCA

SCA (36)
SC (36+10)
BUFF (SC)
FMTNMB (SC+1)
PPNT (SC+2)
SECTOR (SC+3)
RELADR (SC+4)
SAVE (SC+5)
TEMP (SC+6)
CRTSTS (SC+7)
FFMT (FMTNMB)
LFMT (SC+8)
ONE (SC+9)
LUFLAG (SC+10)
CALRPI (SCA+6)
FIELD (SC+11)
LMTLNG (SC+12)
SIGN (TEMP)

d. GENERAL

FW (3)
MOT (5)
EREQST (8)

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME ___PFM - Print Format___   Level 06

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    GETBUF
    SETMOV
    CLCDWA
    ENTSR
    EXTSR
    SUPRW
    GETBIN
    GETCOM
    GETALP
    GETCORE
    REDSEC
    REDDWA
    RELFOR
    CAM
    SV
    REJSUP
    GFNAME
    FMTPAR
    CONCAN

4. DATA AREAS USED AND SYSTEM TABLES

    ATT
    SST
    EXT
    FBA
    FMT
    INTERROGATE BUFFER

5. INTERNAL SUBROUTINES

CNTR.   Inserts counter number in corresponding place in output
                     buffer.
        On entry -   A contains 8 LSB sixth word of entry of current field

TRANSF.   Transfers header to output buffer.
        On entry -   Q contains index of header (1, 2, 3).

SUBR.   Prints output buffer.
        On entry -   Length (gets value in TRANSF) is length of printed line.

BLNKS   Fills interrogate buffer with blanks

CRTCHK   If output unit is CRT, checks for end of screen condition.
                     In case it occurs, it clears screen.
                     Checks for cancel (SX) command.

MOVLIM   Inserts LIMITS parameters in output buffer (converts from
                     BCD form to ASCII).
        On entry -   LMTLNG contains number of words in LIMITS parameter.

ASCII   Converts numbers in binary form to ASCII and inserts in
                     output buffer.

        On entry -   A contains number in binary form.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $33F_{16}$ = 831 words of program.
    99 words of buffer.

2. RELOCATION TYPE

    Run anywhere, disk resident.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME_____PFT - Print Formats from Tape_____Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   The program is divided into two main parts. The first part (PART1)
   prints the list of the formats existing on tape, and the second part
   (PART2) prints the format (or formats) specifications from tape.

   There are two entry points to this program:

   a. ADC PFT - PFTHDR

   b. ADC PFTZ - PFTHDR

   The first one is an entry point to the beginning of the program used for
   both parts of the program.

   The second entry point is used only in PART2, where the program exits to
   another supervisor program PFM (print format) and returns to PFT.

2. DETAILED DESCRIPTION

   In order to understand this program, we should be familiar with tape's
   structure, for detailed description of the format's dump tape (see the
   DMPFMT-IMS)

   Functional Description

   The entered command name and format numbers, if they appear, are checked
   not to contain any input errors. Format numbers must be positive nonzero
   integers, arranged in increasing order and not exceeding the highest
   format number existing in the system.

    Tape is rewound and checked to have the rightheader DMPF containing
   date and last block size used for formats buffer. If tape is not identified
   as the proper one, then a wrong tape message is sent to supervisor interrogate
   buffer, 24 words size is requested through GETBUF.

   Request is made to CAM for two buffers: 34 words for the directory to
   tape buffer and formats big buffer, according to last printed on tape
   up to 16 sectors long, when received, the directory and the formats are
   read into suitable buffers.

   According to the entered command program continues execution in either
   PART1 or PART2.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___PFT - Print Formats from Tape_____Level 06

PART1

A list of formats existing on tape was requested. In this part only the
directory content is checked and format number is transferred to the
interrogate buffer, when full (room for six format numbers) it is printed
on TTY or displayed on CRT.

The directory is searched further on and the interrogate buffer is refilled.

34 words are scanned in the directory unless one of the words is a zero
word. A zero word signifies the end of the directory if there are less
than 16 formats in Big buffer.

When end of directory, advance is made to the next directory.

If tape mark is scanned while reading the directory it means end of
significant data on tape. The nonempty interrogate buffer is printed
on TTY or displayed on CRT and program exits to SV.

PART2 - Format Specifications were Requested

Directory is checked to contain the requested format numbers if both
requested numbers are smaller than those on the tape, the conclusion is that
the requested formats are not on tape and program exits to -SV- if both
numbers are greater than the requested one advance is made to the next
directory. When format is found in the directory, the program calculates
the distance from the beginning of the formats big buffer to the beginning
of the specified format (in words).

Address is saved and the program calls another disk resident program PFM
which analyzes the formats structure and prints it on TTY or displays it
on CRT.

PFM, returns to PFT and the latter checks whether the format analyzed
and displayed was the last requested. If not, it searches the directory
further on and if needed advance is made to the next directory.

As in PART1, tape mark indicates the end of significant data and program
exits to -SV.

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME___`PFT - Print Formats from Tape`_____ Level 06

INTERFACE

1. ENTRY METHOD

   The supervisor enters the following command:

   PFT $\left\{ \begin{array}{l} \text{,nnnn (1)} \\ \text{,nnnn (1), nnnn (2)} \\ \quad\text{all} \end{array} \right\}$ CR

   where: nnnn (1) - first format number
               nnnn (2) - last format number

   When no format number is specified program will print a list of formats
   numbers existing on tape.

   When only one format number is specified the suitable format will be
   analyzed.

   When two formats numbers are specified, all the formats in the range
   of nnnn (2) - nnnn (1) will be analyzed.

2. INPUT CONDITIONS

   Requested format numbers, nnnn (1) and nnnn (2) must be nonzero positive
   integers. Last format number can be greater or equal to the first format
   number nnnn (2) $\geqslant$ nnnn (1). Neither nnnn (1) nor nnnn (2) can be greater
   than the highest format number in the system.

3. EXIT METHODS

   The program can be terminated by one of the following methods:

   a. Error Exit - When an error is detected in the input parameters, in the
      logic structure of the request or if tape is not identified as the
      requested one, program exits to REJSUP with error message.

      Another kind of error may be obtained during tape operations (read,
      advance or rewind).

   b. Return to Caller - When no error occurs and operation is completed
      program exits to -SV.

      Program execution can be terminated manually by pressing MI on TTY
      and typing SX or on CRT hitting the cancel key.

4. OUTPUT PARAMETERS

    a. Program's Error Messages

        1) INVALID COMMAND FORMAT - When comma is not following commands name
           or EOR is missing at the end of the entered command.

        2) INVALID FORMAT NUMBER - When requested format numbers do not fit
           the description in Interface Section, 2 (Input Conditions).

        3) TAPE I/O ERROR - When something is damaged in the tape operation
           (read, advance or rewind).

        4) END OF TAPE

        5) WRONG TAPE - When tape is not identified as the requested one.

        6) MEMORY FULL - RETRY - When CAM fails to get the requested word-size
           buffers.

        7) FUNCTION TERMINATED - If MI and SX are pressed during program
           execution function is terminated.

    b. Completion Message

        There are two types of headers. If a list of format numbers is requested
        the header form will be:

            FORMATS SPECIFIED ON TAPE FROM MMDDYY ARE
            when: MM - month number
                   DD - day
                   YY - year
        in this case formats numbers will appear up to six numbers in a line.

        If format's specifications were requested the header form will be as in PFM.

            FORMAT= STATUS= REC-LNG= FRV= RESEQ=
            NAME   T CH K R V A Z E RSQ F OC SP IFC 2FC L

        when requested format is not found on tape the completion message
        will be:

            REQUESTED FORMATS NOT ON TAPE

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. COMMUNICATION AREA EQUATES - used by PFT

        SUPCOM (36), SC (SUPCOM+10)
        CALRPI (SUPCOM+6) - Caller program index in this case saves the PFT index.
        PPNT (SC+0) - Pointer in print buffer (the interrogate buffer).
        FMTI (SC+1) - First format number zero if a list of format numbers
                     was requested.
        FMT2 (SC+2) - Last format number.
        FMTPN (SC+3) - Save current format number.
        LSTBLK (SC+4) - Block size in words used for last dump received.
        MON (SC+5) - Save month number, from tape's header.
        DAY (SC+6) - Save day received from tape's header.
        CRTSTS (SC+7) - Saves CRT status after displaying the interrogate buffer.
        BIG (SC+8) - Big buffer's backward pointer.
        RELBIG (SC+9) - Relative address in big buffer.
        BUFSIZ (SC+13) - Saves the buffer size requested from CAM.
        TEMP (SC+14) - Flag for displaying header, and in the SERCOR
                     subroutine saves MAXWT.
        YEAR (SC+15) - Saves year, receive from the tape's header.
        FMTPNT (SC+15) - Used in PART2 for fixing the format base address before
                     exit to PFM.
        DIR (SC+16) - Back words pointer in the directory buffer.
        RELDIR (SC+17) - Relative address in the directory buffer.

    b. COMMUNICATION AREA EQUATES - used by PFM

        FIELD (SC+11)
        SAVE (SC+5)
        LMTLNG (SC+12)
        LUFLAG (SC+10)
        CALRPI (SCHA+6)
        SIGN (SC+6)
        TEMP (SIGN)

2. MACROS USED

    None

3.   PROGRAM DEPENDENCIES

     a.   ENTRY POINTS

         1) ADC   PFT   - PFTHDR
         2) ADC   PFTZ - PFTHDR

     b.   EXTERNAL LIST

         1) SETMOV - Entry in ENEXSR, enables moving of disk resident program.

         2) GETBUF - Entry in PIO get buffer subroutine.

         3) CONCAN - Core resident, checks for cont/cancel reply at end of
                 screen.

         4) GETALP ($9B) - Entry in SCANV, retrieve the comma following
                 command.

         5) GETCOM ($9C) - Entry in SCANV, retrieve the EOR following command.

         6) GETBIN ($99) - Entry in SCANV, retrieve the formats nos.

         7) REJSUP ($B2) - Entry in SUPER, exit when an error occurs.

         8) TAPIO ($9A) - Entry in PIO, performs tape motion, reads and writes
                 (from and to specified location.)

         9) SUPRW ($97) - Entry in PIO called to read or write from or to suitable
                 logical unit.

       10) RELFOR ($AD) - Entry in GFCORE, release the requested format.

       11) SV ($B1) - Entry in SUPER, exit when command is accomplished.

       12) CAM ($AE) - Entry in CAM, request core space for program buffers.

       13) WAIT ($B0) - Entry in ENEXSR, used to delay function for core cycle.

       14) ENTPRG ($9E) - Entry in ENTPRG, disk resident programs are
                   accessed vice ENTPRG used to call PFM

       15) EXDFUN ($9F) - Entry in ENTPRG, called to release a disk resident
                   program releases PFT when PFM is entered and vice
                   versa.

4. DATA AREAS AND SYSTEM TABLES USED

   PFT - none
   PFM - LFD
         FMT

5. INTERNAL SUBROUTINE

   SEROR - requests core for the program buffers.
           Input parameters: Register Q - No. of requested words
                                            excluding header.

   IONOF - Sets and clears I/O bit.
           Input parameters: Register Q - Back words pointer of the buffer.

   TRANSF - Transfers requested message to the interrogate buffer.
            Input parameters: Registers Q - Index of the message line.

   SUBR - Prints the message on the suitable device.
          Input parameters: Register Q - Index of the message length.

   BLNKS - Checks blanks in the interrogate buffer.

   CRTCHK - Checks CRT after displaying the interrogate line.

   CHKTAP - Checks tape after any tape operation:
            Read advance or rewind.
            INPUT parameters: after coming from TAPIO Register A tape's status.

PHYSICAL REQUIREMENTS

1. MEMORY SPACE REQUIREMENT

   IEI_{16} = 481 words.

2. RELOCATION TYPE

   Disk resident and run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME_____ PIO - Physical Input/Output _____ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    PIO is a collection of physical I/O subroutines. There are 21 program
    entry points. Entry is by return jump followed by a parameter list of
    0 to 5 words.

    a. WSS writes a statistics record on the Cyberdata Statistics File.

    b. READP reads the previous record.

    c. WRITEC writes the current record to disk.

    d. REWRIT rewrites a record after corrections are made.

    e. FRZATT writes current ATT on the second sector of the first track
        of batch.

    f. WRTATT writes from the current ATT the disk copy of the ATT and saves
        the clock value on the second sector of the first track of the batch.
        Note that this entry is used to write check points in order to be
        able to recover after a power failure.

    g. READN reads the next record.

    h. SETNFM sets the no forward motion indicator in verify mode and displays
        a 'B2' end of data error in read mode when read next record encounters
        the last record.

    i. REDWRD reads the disk by word addressing.

    j. WRTWRD writes the disk by word addressing.

    k. CLCDWA calculates a disk word address given a sector and displacement
        within the sector.

    l. GETSTR indicates a read request from the key station into the interrogate
        buffer.

    m. GETBUF calls CAM to get an interrogate buffer for a terminal.

n. REDSEC reads the disk by sector addressing.

o. WRTSEC writes the disk by sector addressing.

p. SUPRW reads from the supervisor input unit or writes to the supervisor list unit.

q. TAPIO handles mag tape requests from Cyberdata routines that are specifically designed to use mag tapes.

r. MOVREC moves a block from one core address to another.

s. ABSADD sets an internal flag to indicate that the next entry to PIO will be of absolute addressing.

t. ASK17 called by ASKERR to terminate the recovery if core is not available.

u. CONVRS displays conversational responses for supervisor functions, and is used by LBL1.

2. DETAILED DESCRIPTION

PIO has a common exit to issue an I/O request (COM6). Before the I/O request is made, a check is made for code conversion. (CHKCVT) calls the common entry subroutine. If the logical unit in the I/O request is not the supervisory tape unit, exit is made via the common exit subroutine. If the request is a mag tape one, the request code is checked to see if conversion is necessary. If code conversion is necessary, PIO moves the request parameters to a temporary storage area (because the ATT parameter list area is used by ENTPRG) and calls the appropriate conversion module via ENTPRG. When the code conversion is complete, the original parameter list is restored and exit is via the common exit subroutine. (Conversions for ASCII and EBCDIC are available, and four hooks are in for user code conversions.) A standard completion address (COMPL) and the required request code and priority (increased or decreased according to SST/RPINCR) are stored in the ATT parameter list. The requesting terminal is function locked (ATT/SLOCK/14), an indirect request is issued, and exit is made to the dispatcher to wait for the I/O to complete. (All Cyberdata standard I/O calls are made via this indirect request through the ATT.)

PIO also has a standard I/O completion routine (COMPL) to restore I to
the ATT base, check for code conversion (CHKCVT), clear the I/O bit, and
check for errors. If no errors occurred, the current field format address
and current ATT are set in SST. Function lock is cleared in the ATT.
Q is set to the error indicator (V field) and A is set to the device status
(PDT/ESTAT2). Control is returned to the caller via the common exit subroutine
EXTSR. If the error was not a mass memory one, the normal exit is taken with
the error passed to the caller in Q. If the error was a mass memory error,
clear function lock and cause TTY time-out in case the supervisor input
is in process. If the mass memory error was during recovery of another
disk error, the following messages are printed on the system list device:

<div align="center">'DISK ERROR DURING RECOVERY'
'STATION nn LOCKED'</div>

The station is set to function lock and exit is to the dispatcher. If the
disk error was not during recovery and the disk is not ready, 'DISK n NOT
READY' is printed on the system list device. Otherwise, the faulty sector
address is calculated and saved in ATT/ATEMP. The station is put to
supervisory lock and the stack is cleared. If a disk resident routine was
executed on the station it is released.

The disk part of the recovery program is loaded via ENTPRG and executed.
If not sufficient core was available for this routine (DSK17) the
following messages are printed on the system list device:

<div align="center">'DISK I/O ERROR. NO CORE FOR RECOVERY PGM'
'STATION nn LOCKED'</div>

The station is set to function lock and exit is to the dispatcher.

a. WSS calls the common entry subroutine ENTSR. It calculates the amount
    of statistics file space used; if more than 90% is used, it prints a
    warning message to the system list device. If there is statistics file
    disk space remaining, WSS calculates a disk word address, sets up the
    unformatted write parameters in the ATT (word 65 on), and jumps to the
    PIO common exit subroutine for I/O calls.

b. READP calls the common entry subroutine ENTSR. It finds the previous
    record disk address in the terminal record buffer (TRB/PREVDA) and
    stores it in ATT/CURDA and ATT/CURDA + 1 (the current record address)
    and in ATT/RQ + MMA and ATT/RQ + MMA + 1 (the I/O request parameter
    list disk address). The TRB length (minus pointers) and unformatted
    read core are set in ATT. The I/O bit is set on the TRB and the logical
    unit for this batch is stored in ATT/RQ + LU. Then READP goes to the
    common exit to issue an I/O request (COM6). If there is no previous
    record, a B1 error (beginning of data) is generated.

<div align="center">PAGE NO. __549__</div>

c. WRITEC calls the common entry subroutine. The next disk address of the
    ATT is moved to the current disk address (WRTCOM). The ATT current
    disk address is stored in ATT/RQ+MMA and ATT/RQ + MMA + 1. The record
    length plus a 5 word header is stored in ATT/RQ + N. An unformatted
    write is set in ATT/RQ, the beginning address of the buffer is stored
    in ATT/RQ + S and the I/O bit is set for it. The batch logical unit is
    set in ATT/RQ + LU and COM6 (PIO common exit to issue I/O) is executed.

d. REWRIT calls the common entry subroutine. It clears ATT/SW1/13 (rewrite
    bit) and goes to the WRTCOM label in WRITEC (described above).

e. FRZATT calls the common entry subroutine. (RWATT) It sets a request
    of formatted write, sets the ATT base as the buffer address and clears
    the I/O clearing bit flag ATT/IODDR. It calculates the ATT sector
    address and stores it in ATT/RQ + MMA and ATT/RQ + MMA + 1. It sets the
    request for writing 64 locations, sets the logical unit for this batch
    and executes the PIO common exit to issue I/O.

f. WRTATT calls the common entry subroutine, saves the clock value in ATT/STRC
    and calls the common entry subroutine again in order to clear the ATT/STRC
    on the completion of the request. (This is done for the reason of not
    losing statistical information on a power-fail condition). The program
    continues at RWATT label in FRZATT (described above).

g. READN calls the common entry subroutine. If there is another record,
    READN moves the next record disk address into ATT/CURDA, ATT/CURDA + 1
    and ATT/RQ + MMA, ATT/RQ + MMA + 1. It sets the record length and
    unformatted read code, beginning address and batch logical unit in the
    ATT parameter list. It sets the buffer I/O bit and executes the PIO
    common exit to issue I/O (COM6).

h. SETNFM. If there is no record, SETNFM sets the relative format field
    address (ATT/RFA) to 4 and clears the current field character count.
    If the rewrite record flag (ATT/SW1/13) is set, it rewrites the record
    (REWRIT). The ATT address of the last record entered is stored as
    the TRB previous record. If the terminal is in read mode, a B2 (end of
    data) error is displayed. If the terminal is in verify mode, ATT/SW2/11
    (no forward motion) and ATT/CMODE/14 (IDK flag) bits are set, ATT/SW1 and
    record count are cleared. The screen is cleared, an ATT check point is
    written and exit is to CLRSR.

i. REDWRD sets up a unformatted read request in the ATT parameter list
    according to the caller's parameter string, adjusts the buffer for
    relative addressing (if necessary), and clears ATT/IOADDR (I/O bit
    clearing needed flag). It increments the return address past the
    parameter string before calling the common I/O entry subroutine ENTSR.
    Then it picks up the logical unit for the read and executes the PIO
    common exit to issue I/O.

j. WRTWRD is the same as REDWRD except for using an unformatted write request.

k. CLCDWA multiplies the sector address by 96 (words/sector) and adds the
    word displacement to it. The resulting word address is in QA with bit 15
    of A not used.

l. GETSTR calls the common I/O entry routine to save its return address in
    the ATT exits stack. It calls GETBUF to get core for an interrogate
    buffer, sets ATT/CMODE/15 (interrogate mode flag) and exits to the
    dispatcher.

m. GETBUF calls the common entry subroutine. It checks to see if there is
    already an interrogate buffer allocated to the terminal, and if so, if
    that buffer is large enough for the current request. If there is an
    interrogate buffer that is too small, GETBUF releases it (RELFOR) and
    allocates a new one by calling CAM. If a new buffer is needed, CAM is
    called and GETBUF waits (WAIT) indefinitely until the core request
    can be satisfied. When the buffer is obtained, GETBUF sets the backward
    pointer to ATT/IRB, clears the character count, and sets all words in
    the buffer to minus zero. It clears the I/O bit (IRB/1/15) and completion
    code (IRB/2/7-15) of the interrogate request buffer and exits to the
    common exit subroutine.

n. REDSEC is the same as REDWRD except that the request is a format read.

o. WRTSEC is the same as WRTWRD except that the request is a format write.

p. SUPRW transfers the request code, logical unit, ASCII/binary bit, buffer
    address, and buffer length or motion code to the ATT parameter list. It
    adjusts the caller's return address according to the number of parameters
    in the calling sequence and calls the common entry subroutine to save
    the return address in the ATT exits stack. If the interrogate buffer is
    being used and a read request is initiated, SUPRW calls GETBUF to
    determine if the interrogate buffer is long enough. If the request was
    not from a Cyberdata terminal, the interrogate buffer I/O bit is set, if
    necessary, and SUPRW jumps to the PIO common exit to issue an I/O request.
    If a read request was from a Cyberdata terminal, a read is initiated by
    GETSTR. If a supervisory write or motion request was issued, the terminal
    is function locked to stop input, the screen is cleared for a motion
    request, a line is displayed for a write request and the end of display
    area is indicated. Control is then passed to the common I/O completion
    section of PIO where SST values are restored before returning control to the
    caller via EXTSR.

q. TAPIO first checks to see if code definition (ASCII or BCD) is required
    and adjusts the request code accordingly. Then it checks ATT/SUPCOM + 2
    to see if a supervisory tape logical unit has been set. If not, it
    returns to the caller with error completion. If the tape logical unit is
    set and the request is not a motion request, ATT/BCOUNT is incremented and
    TAPIO jumps to label ABMODE in SUPRW to set up the I/O parameter list.
    If the request is a motion request, the motion parameter (bit 15) is
    checked for a repeat indicator. The block count is adjusted according
    to the type of motion request and the repeat increment before TAPIO
    jumps to ABMODE (as above).

r. MOVREC moves a block from one part of core to another. The addresses
    specified in the calling sequence can be either relative or absolute
    and control is returned directly to the caller.

s. ABSADD returns directly to the caller, the value in the entry point when
    it is not zero specifies for the PIO subroutine that the following entry
    to PIO is of a core resident routine with absolute buffer address (this
    is needed to distinguish from the upper part of the system when
    executing Cyberdata in more than 32k).

t. DSK17 is entered from DSKERR if no core is available (see in Detailed Description label DSK17).

u. CONVRS first saves the return address in the ATT exits stack via ENTSR. The number of words last displayed is converted to character count and saved in ATT/ATEMP + 1. The number of words to be saved is stored locally. If not terminal zero, the display mode is changed to Interrogate. A read request is issued via SUPRW. If the terminal is zero, exit is via EXTSR. Otherwise, the display mode is restored to data, and if no characters were input exit is via EXTSR. If characters were received, a check is made to see whether the previous message plus this response will fit on one line. If so, parameters are prepared to display in this layout. If the two messages do not fit on the same line, the new response is displayed on the next line via SUPRW. Exit is via EXTSR.

INTERFACE

1. ENTRY METHOD

    RTJ followed by Parameter list (0-5 words).
    I = ATT base address.

2. INPUT PARAMETERS OR CONDITIONS

    a. WSS: Words 34-49 of the TRB are set up for the statistics block.

    b. READP: Disk address of the record to be read is saved in PRVDA of the TRB, logical unit in the ATT must be set up correctly.

    c. WRITEC: The record in the TRB will be written on the disk specified in ATT/LU at the address specified in ATT/NXTDA,NXTDA + 1.

    d. REWRIT: The record in the TRB will be written on the disk specified in ATT/LU at the address specified in ATT/CURDA,CURDA + 1.

    e. FRZATT: ATT will be saved on the second sector of the track specified in ATT/FTOB.

    f. WRTATT: As in FRZATT.

    g. READN: Disk address of the record tobe read is saved in NEXTDA in the TRB. ATT/LU must be set up to the appropriate disk unit.

    h. SETNFM: No parameters.

    i. REDWRD: Buffer address.

j. WRDWRD: MSB of disk address.
             LSB of disk address.
             No. of words.
             Logical unit.

k. CLCDWA: A = Sector
             Q = Word displacement

l. GETSTR: A = Number of characters to be read in (including REL).

m. GETBUF: A = Number of characters.

n. REDSEC: Buffer address.

o. WRTSEC: Sector address.
             Number of sectors/words (value is in one's complement if it
             specifies number of words).
             Logical unit.

p. SUPRW: For data transfer: Buffer address = 0 = Interrogate Buffer
                                 Number of words
                                 Q = 0 for READ   ASCII
                                       1 for WRITE   ASCII
                                       2 for FREAD   ASCII
                                       3 for FWRITE ASCII
             For motion: Motion Code (2000 - page eject/clear screen)
                                 Q = 5

q. TAPIO: For data transfer: Buffer address = (0 = Interrogate buffer
                                 Number of words
                                 Q = 0 for READ   ASCII
                                       1 for WRITE   ASCII
                                       2 for FREAD   ASCII
                                       3 for FWRITE ASCII
                                       4 for MOTION (Input Unit)
                                       5 for MOTION (Output Unit)
                                       8 for READ   Binary
                                       9 for WRITE   Binary
                                       A for FREAD   Binary
                                       B for FWRITE Binary
                                      16 for READ   SPECIFIED CODE
                                      17 for WRITE   SPECIFIED CODE
                                      18 for FREAD   SPECIFIED CODE
                                      19 for FWRITE SPECIFIED CODE

For motion: Motion Code

          1 - Backspace record
          2 - Write end-of-file
          3 - Rewind
          4 - Rewind and unload
          5 - Skip one file forward
          6 - Skip one file backward
          7 - Advance one record

If a data transfer with a specified code is required.

        $A = 0$ for BCD
            1 for EBCDIC
            2 for ASCII
            3 for conversion routine (1)
            4 for conversion routine (2)
            5 for conversion routine (3)
            6 for conversion routine (4)

r. MOVREC: ADC "FROM" ADDRESS (ABS/REL)
           ADC "TO" ADDRESS (ABS/REL)
           NUM Number of words to move

s. ABSADD: No parameters

t. DSK17:   No parameters

u. CONVRS: A = Length of response to be requested.
           Q = Length of last message displayed.

3. EXIT METHOD

Return to caller at address immediately following parameter list.

4. OUTPUT PARAMETERS OR CONDITIONS

   a. WSS: Statistics record written to disk.

   b. READP: Previous record read into TRB.

   c. WRITEC: Current record written to disk.

   d. REWRIT: Current record rewritten to disk.

   e. FRZATT: ATT is written on 2nd sector of 1st track of batch.

AA5641   PRINTED IN THE USA

f. WRTATT: ATT written on 2nd sector of 1st track of batch.

g. READN: Next record is read.

h. SETNFM: B2 error displayed in read mode.
             No forward motion indicator set in verify mode.

i. REDWRD: Disk read (word addressable).

j. WRTWRD: Disk written (word addressable).

k. CLCDWA: Q A = disk word address.

l. GETSTR: Exits to dispatcher. Return to caller by MPC when:

             1. More than requested number of characters is read.
             2. Release key depressed or
             3. Cancel key depressed.

m. GETBUF: ATT/IRB set to buffer address.

n. REDSEC: Disk sector(s) read.

o. WRTSEC: Disk sector(s) written.

p. SUPRW: ⎫
           ⎬ Appropriate action taken.
q. TAPIO: ⎭

r. MOVREC: Block of core moved.

s. ABSADD: Absolute address flag is set # zero.

t. DSK17:   No parameters.

u. CONVRS: Input from supervisor.
             Display of input response if not terminal zero.

Note that ATT temporaries are not destroyed during the PIO routines
unless a request was made to TAPIO.

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME ___PIO - Physical Input/Output___   ___Level 06___

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SIX ($44)
        TEN ($46)
        SST ($47)
        MONITR ($8C)
        CLCDW ($90)
        ENTSR ($91)
        EXTSR ($92)
        CLRSR ($93)
        ERROR ($96)
        SPRW ($97)
        ENTPRG ($9E)
        EXDFUN ($9F)
        DISPLY ($A0)
        RWRITE ($A4)
        WATT ($AC)
        RELFOR ($AD)
        CAM ($AE)
        WAIT ($B0)
        SV ($B1)
        DISPCH ($EA)

    b. SST

        EXTSST (0)
        ATTB (10)
        CATT (11)
        CFFA (12)
        RPINCR (SST + 13)
        IBL (14)
        TIQ (15)
        CLOCKS (17)
        STEMP (20)
        TEMP1 (20)
        TEMP2 (21)

c. EST

    SFDA (5)
    SFDE (6)
    SFRC (7)
    LUND1 (25)

d. CONSTANTS

    I/O REQ:
      C (1)
      T (2)
      LU (3)
      N (4)
      S (5)
      MMA (12)
      ESTAT2 (12)
      UR (0)
      UW (1)
      FR (2)
      FW (3)
      CB2 (7)
      STACKL (8)

e. ATT

    FTOB (1)
    CMODE (2)
    CURCH (2)
    RCNT (4)
    ATRB (6)
    FBA (7)
    CURDA (9)
    NDA (11)
    RFA (14)
    BMODE (15)
    SW1 (19)
    SW2 (20)
    FUNIDX (21)
    IRB (27)
    TOPT (33)
    STRC (34)
    LUN (35)
    SUPCOM (36)
    BCOUNT (SUPCOM + 5)
    DELAYC (46)
    RQ (65)
    IOADDR (RQ - 1)
    ATEMP (73)
    ENTRY (78)

f. TRB

  PREVDA (4)
  NEXTDA (5)
  NXTPRV (6)

g. TOPT

  CURSRO (3)

h. EXTERNALS

  LOG1A contains addresses of the PHSTAB's for all the logical units.
  CLRALL⎫
  DLINE ⎪
  DEOREV ⎪
  REGEN ⎬parameters for display.
  SETDIS ⎪
  DSPDIR ⎭
  CCKEND entry point in DISPLY to check end of screen.
  SUPTHD entry point in SUPER (thread word of the read request).
  CARLIN characters per line.

2. MACROS USED

  FWRITE
  INDIR

3. PROGRAM DEPENDENCIES

  MONITR

	ENTSR
	EXTSR
ENEXSR	CLRSR
	WAIT
ENTPRG	ENTPRG
	EXDFUN
GSE	ERROR
DISPLY	DISPLY
	CCKEND
GFCORE	RELFOR
CAM	CAM

```
NDISP    DISPCH

         CLCDW
         SPRW
P10      RWRITE
         WATT

         SV
SUPER    SUPTHD
```

4. DATA AREAS USED

Physical device table (LOG1A)

ATT

SST

EST

TRB

TOPT

IRB

PRD

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $40D = 1037_{10}$.

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___PJB - Print Job_____ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    To process the PJB command, whose function is to print data from LJD.
    Two formats of the command are possible.

    a. PJB
        Prints all existing jobs in system.

    b. PJB,xxxxxx
        Gives information on the job named xxxxxx.

2. DETAILED DESCRIPTION

    The command is scanned to check format and is processed accordingly.
    In case of syntax error return is made to supervisor mode with the error
    message "Invalid command format".

    a. PJB,xxxxxx format

        After processing the job name (xxxxxx), GETLJD is invoked (with 1 in
        Q as parameter) to get the fourth and fifth words in the corresponding
        LJD entry. (Into MXRC and VR.) If the job name is not found (GETLJD
        returns Q = 0) return is made to supervisor mode with the error message
        "Undefined Job name".

        If the job is found the information is printed under a corresponding
        header.

        Return is made to supervisor mode.

    b. PJB format.

        The whole LJD is brought into core, sector by sector. If no core is
        available (after 100 cyber-cycles) return is made to supervisor mode
        with the message "Memory full, retry".

        When there is core available, the first sector is brought into core
        and a header with number of jobs in system is printed. In case there
        are no active jobs in system return is made to supervisor mode; no
        message is printed.

After the header has been printed, the program checks if "Cancel" command
was entered, in which case it releases core and returns to supervisor
mode with message "Function terminated" (same for each line). Otherwise,
it prints the job names out of the LJD, leaving two blanks between two
successive names, and six names per line.

In case the output is on the CRT, the program waits for a "Cancel" or
"Continue" command upon end of page.

INTERFACE

1. ENTRY METHOD

    PJB is loaded by ENTPRG. Execution begins at only entry point PJB.

2. INPUT PARAMETER

    PJB command in either of two possible formats.

3. EXIT METHOD

    a. Completion Exit

        JMP - (SV)

    b. Cancel Exit

        ENA 39
        JMP - (REJSUP)

    c. Error Exit (syntax or job name error)

        ENA 9
        JMP - (REJSUP)

    d. No Core Exit

        ENA 11
        JMP - (REJSUP)

4. OUTPUT CONDITION

    See Cyberdata System Reference.
    I = current ATT address.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___PJB - Print Job_____Level 06

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

       LPMASK ($2)
       ONE ($3)
       NZERO ($12)
       ONEBIT ($23)
       ZEROBIT ($33)
       TEN ($46)
       WAIT ($B0)
       GETCOM ($96)
       REDSEC ($96)
       SUPRW ($97)
       REJSUP ($B2)
       RELFOR ($AD)
       ENTSR ($91)
       EXTSR ($92)
       CAM ($AE)
       GETSTR ($A2)

    b. GENERAL

       FW (3)
       MAXWT (100)
       MT (9)
       MOT (5)
       MRL (19)
       JN (1)
       E ($45)
       F ($46)
       R ($52)
       U ($55)
       V ($56)

c. SST, ATT

    SST ($47)
    ATTB (10)
    CATT (11)
    STEMP (20)
    LJDDA (8)
    VO1 (13)
    VO2 (14)
    VO3 (15)
    EOC (17)
    TRMNT (23)
    OUTLU (37)

d. SCA

    SUPCOM (36)
    SC (SUPCOM)
    COUNT (SC+10+0)
    INTRM (SC+10+1)
    INTRM1 (SC+10+2)
    INTRM2 (SC+10+3)
    TEMP (SC+10+4)
    BLKCNT (SC+10+7)
    PTRQ (SC+10+8)
    LNCNT (SC+10+9)
    CELL (SC+10+10)
    LJDB (SC+10+12)
    JOBNM (SC+10+16)
    TEMP1 (SC+10+17)

2. INTERNAL ROUTINES

    WTCOR - WAIT CORE
        On entry - Q number of cyber-cycles to wait.
            exit - via REJSUP in case no core is available.
    SECTOR - Read one sector in core
        On entry - LJDB+ATT contains address of buffer.
    FILLB - Fills interrogate buffer with blanks.
    BCDASC - Converts 3 digit BCD number into ASCII form.

The strategy used for the recording code is used again for the blocking
mode (W0,B14-15).

Next parameter retrieved is the block length (W1); it is converted to
ASCII (after doubling it) and stored as output by the BINASC subroutine.
Same procedure for the record length (W2).

The padding type is obtained (MTFBUF W0-B9), together with the padding
character. If everything is zero, 'N' (no padding) is stored in the
interrogate buffer. Otherwise, 'A' or 'B' is stored, according to the
'P' bit. In that case either the padding ASCII character or the representation
of its decimal value are stored in consecutive locations in the interrogate
buffer.

The own code number (MTF-W0-B9-13) is read, converted to ASCII representation
and stored in the interrogate buffer (by the BINASC routine). The same
procedure is repeated for the file label number (W1,B11-15). The ASCII
character 'S' or 'P' is inserted in the output line depending on bit 12
word 3 of MTF. And lastly, the batch grouping factor (W3,B1-11) is
converted to ASCII and sent to the output line by the BINASC routine. The
resulting interrogate line is then printed by the PRINT routine, after saving
the return address through ENTSR. Also, end-of-screen indication is stored
in CRTSTS for further cleaning of display.

After all the word was done on one MTF, MTFN1 is increased and check if
equals MTFN2, then program exits to SV. Otherwise, control passes to LOOP1
and the word starts again, on the next MTF.

CRTCHK - Is a routine to check the screen status (CRTSTS). If end-of-screen
was detected, the CONCAN subroutine detects the depression of continue or
cancel keys. If cancel, a message 'Function terminated' is sent (after
clearing the screen) and exit is to REJSUP.

BLANKS - A subroutine to blank out the interrogate buffer.

BINASC - Converts binary numbers passed in A to their equivalent ASCII
representation and stores them in the interrogate buffer in consecutive
locations. At entry, STEMP points to the last word to contain the least
significant digits.

LIST - All the active formats. A request is made for 95 words of core.
If not available, the command is rejected. If core obtained, the backwards
pointer (in the interrogate buffer header) is set to point to BCKP and the

buffer itself is blanked (by BLANKS). Then, after calculating the disk
address corresponding to MTF no. 0, 95 words are read from the disk to the
interrogate buffer. The I/O bit is set before reading and cleared after it.
Then, all the formats are handled in the same manner. MTFN contains the
current format number. The word is ended when MTFN exceeds 255. If current
format is not active, MTFN is just increased and a check is made to see if
all the interrogate buffer was inspected (at CONT). In that case, a new block
of 95 words is read from disk (at NEWSC).

If current format is active then either the constant header (HEAD2) is moved
to the interrogate buffer (at PRHD2) and printed, or the current line is
printed (at NT). At NT1, a new MTF number is obtained, converted to ASCII
(by BINASC) and stored in the output line. Pointers to the 95-word block
are updated (at CONT) and the next MTF is looked for being active.

Before exit the user count in the interrogate buffer is set to 0.

INTERFACE

1. ENTRY METHOD

    PMT is entered from DSKSUP via ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

    (I) = ATT base address.

3. EXIT METHOD

    If any errors detected, exit to REJSUP with a corresponding error message
    index. If no errors, exit to SV.

4. OUTPUT PARAMETERS

    None. (I) is unchanged.

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ PJB - Print Job _____ Level 06

3. MACROS USED

    None.

4. PROGRAM DEPENDENCIES

    SV
    SUPRW
    REJSUP
    RELFOR
    BD
    ENTSR
    EXTSR
    SETMOV
    CAM
    GETSTR
    GETLJD
    GETBUF
    CONCAN
    MOVREC

5. DATA AREAS AND SYSTEM TABLES USED

    SST
    ATT
    LJD
    SCA
    INTERROGATE BUFFER
    EST

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    179_{16} = 377 words for program.
    99 words for buffer.

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___PMT - Print Mag Tape Format_____ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The command syntax is:

    $$\text{PMT} \quad [\,,\text{aaa} \quad [\,,\text{bbb}]\,]$$

    'aaa' is the first (or only) format number to be printed; 'bbb' is the last
    format number. If neither 'aaa' nor 'bbb' are specified, all the format
    numbers in the system are printed out. If only one format is required,
    'bbb' should not be specified. The PMT program prints out the desired
    mag-tape formats specifications in a fixed form.

2. DETAILED DESCRIPTION

    PMT starts by clearing several temporary variables and making itself
    movable. Then a check is made for the next parameter. If end-of-record
    follows, control passes to LIST and all the active formats are listed.
    If comma follows, the first number 'aaa' is retrieved (at NUM1). Otherwise
    the command is rejected as erroneous.

    If 'aaa' is specified, it is read and stored in MTFN1. The same procedure
    is followed for 'bbb' (stored in MTFN2). Errors arise if either one of
    these is illegal (nonnumeric), comma after 'aaa' but 'bbb' is missing, 'bbb'
    is less than equal to 'aaa', or 'bbb' is greater than 255.

    If no errors, 5 words corresponding to the current format are read from
    disk into MTFBUF (at LOOP1). If that format is not active (bit A/I is 0)
    and is the only format required, an error message is issued. Otherwise,
    the next format is retrieved (TRYNXT). If the current format is active
    then at ANALIS either HEAD1 the format header is moved to the interrogate
    buffer, printed and the screen checked by CRTCHK, or (at CKTAB) the
    various parameters are moved to the interrogate buffer (blanked in any
    case).

    First, the format number is obtained (MTFN1), converted to ASCII by the
    BINSC subroutine, and stored in the interrogate buffer. Next, the
    recording mode is obtained (MTFBUF - W2) and stored as ASCII character
    in the interrogate buffer.

    The recording code (MTFBUF - W2) is transferred to Q and used as an index
    to the table (RCODE) containing the ASCII equivalents of the recording
    code. After transferring it to the interrogate buffer, the over punching
    code is retrieved (W3,B13-14). According to those bits, a corresponding
    letter is entered in the interrogate buffer (Y,N or -).

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME __PMT - Print Mag Tape Format__   Level 06

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ZROBIT ($33)
        NZERO ($12)
        ONEBIT ($23)
        TEN ($46)

    b. SST

        SST ($47)
        MTFADD (42)
        STEMP ($5B)
        EXTSST (0)

    c. TRANSFER VECTOR

        GETALP ($9B)
        GETSTR ($A2)
        SUPRW ($97)
        CAM ($AE)
        REDDWA ($A9)
        GETCOM ($9C)
        REJSUP ($B2)
        EXTSR ($92)
        REDSEC ($A7)
        WRTDWA ($AA)
        GETBIN ($99)
        SV ($B1)
        ENTSR ($91)
        CLCDWA ($90)
        WAIT ($B0)

d. SUPERVISOR COMMUNICATION AREA

    FW (3)
    SUPCOM (36)
    MTFN (38)
    MTFNO (41)
    MAXMTF (44)
    CRTSTS (47)
    MOT (5)
    MTFN1 (36)
    FLAG (39)
    RELADD (42)
    WORDA (45)
    MTFBUF (48)
    MAXWT (100)
    MTFN2 (37)
    BUFF (40)
    MTN (43)
    BCKP (46)

e. ATT

    AINT (27)
    CANC (23)

2. MACROS USED

    None.

3. EXTERNALS

    GETBUF
    PRTMSG
    CONCAN
    SETMOV
    BD

4. DATA AREAS USED

    Interrogate Buffer
    ATT
    MTF
    SST

5. CONSTANTS USED

    A ($41)
    MINUS ($2D)
    V ($56)
    U ($55)
    B ($42)
    Y ($59)
    F ($46)
    S ($53)
    E ($45)
    N ($4E)
    D ($44)
    P ($50)

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $1D6_{16}$ = 470 words.

    Buffer of 95 words for list.

2. RELOCATION TYPE

    Loader relocatable, run anywhere.

3. REENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___ PPR - Print User Programs _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   PPR is a supervisor function to list user data check own code programs
   or user magnetic tape programs of which there are three types:
   own code, tape label and code conversion.

2. DETAILED DESCRIPTION

   a. Section 1

      PPR receives control from DSKSUP after storing the PPR command in
      the interrogate buffer. SETMOV is called to allow the module to be
      moved and the parameters are checked for syntax. For errors a jump
      is made to REJSUP with the appropriate error code in A. DIRTYP
      (ATT/2E) is set depending on the parameters as follows:

Parameters	DIRTYP
0	0
T, 0	1
T, L	2
T, C	3

      Control proceeds to section 2 (NOSUB)*

   b. Section 2 (NOSUB)

      An interrogate buffer of 48 characters is requested for use as a
      print line buffer. CRTSTS (ATT$30) is set to zero to indicate first
      time.

      Depending on the directory or sub-directory required, i.e., value of
      DIRTYP, the starting address and last address + 1 in the PRD are
      calculated and stored in DIRADD and DIRLIM (words $31 and $32 of ATT).
      For own codes DIRLIM is calculated from distance to own code entries
      in PRD/1 and maximum number of own codes in EST/4. The sub directory
      structure of the mag-tape entries in the PRD is as follows. Words
      0-31 are reserved for mag tape own codes, words 32-63 for tape label
      routines and words 64-68 for code conversion programs. A counter
      in DIRIND (ATT$2F) is set to zero and control is passed to section 3
      (ADVANC).

   * Note: names in brackets refer to labels in program listing.

c. Section 3

1) (ADVANC). The counter DIRIND and the current address DIRADD
    are incremented to point to the next entry of this directory
    in the PRD.

2) (NXTPRG). A check is made to see if end of directory has been
    reached i.e. DIRADD = DIRLIM. If not, go to step 3). Otherwise a
    check is made to see if LINIX (ATT/34)= 5 indicating the current
    print line has been output. If not, routine DISLIN is called to
    outprint the line and exit is made to SV.

3) (CHKPRG). If the PRD entry is 7FFF, i.e. does not exist, control
    goes back to step 1). If the entry does exist a check is made on
    CRTSTS for the first time and if not control goes to step 4). If
    first time routine BLANKS is called to initialize the output device
    and clear the display line buffer to blanks. Note also LINIX is
    set to 5 in BLANKS (see step 2)). Routine MOVMSG is called to
    move one of the four following header messages to the display line
    buffer depending on DIRTYP the type of directory requested.

DIRTYP	Header Message
0	Own-code progs.
1	Mag-tape own-code progs.
2	Mag-tape labeling progs.
3	Mag-tape code-conversion progs.

    Routine DISLIN is called to output the header line and routine
    BLANKS called again to reinitialize the display line buffer.

4) (ADVLIN). The address of the interrogate buffer from ATT/$1B
    is increased by value of LINIX (initially) 5 for a display line)
    and stored in SST + STEMP as a pointer for the place in the display
    line buffer to put the next PRD directory entry. This is obtained
    from the PRD using DIRIND, converted to ASCII and placed in the
    display line buffer.

5) (IFDISP). When the value of LINIX reaches a value of 27 this
    indicates the display line buffer is full and it is then output
    and initialized by calls to routines DISLIN and BLANKS. Otherwise
    LINIX is increased by 2 to allow for 1 word of blanks between
    printed entries and control is passed to step 1).

d. SUBROUTINE BLANKS (clear display line buffer to blanks)

1) (BLANKS). As I/O is being done ENTSR is called to put return
    address in ATT stack and SETMOV is called to reset the function
    index relative bit (and incidentally allows module to be moved).

2) (EOLINE). Initialize printer in display line buffer LINIX to 5.
    If CRTSTS = 0 i.e. first line go to step 4). If the output device
    is a CRT and the cursor is at end of screen (when DIRIDX = $C000
    set in DISLIN after each line is output), go to step 3) otherwise
    go to step 5).

3) Call routine CONCAN to wait for continue or cancel key to be
    pressed. On return if A = 0 (i.e. continue) go to step 4). If A = 1
    (cancel) exit from PRD by jump to SV.

4) (MOTION). Sends WEOF motion request with call to SUPRW to either
    clear display or advance page depending on the output device.

5) (MVBLNK). Fill interrogate buffer words 4-27 with blanks. This
    is the display line. Return to caller via a jump to EXTSR.

e. SUBROUTINE MOVMSG (Moves header message to display line buffer)

Gets DIRTYP containing 0, 1, 2, or 3 depending on type of directory
requested. Sets A to point to corresponding header message and Q to
its length. Moves word by word message to display line buffer. Returns
to caller.

f. SUBROUTINE DISLIN (Output one line)

As I/O is being done ENTSR is called to put return address in ATT stack
and SETMOV is called to reset the function index relative bit. SUPRW
is called to send display line in interrogate buffer to the output
device. The CRT status is saved in DIRIDX (ATT/$33) for checking later
in subroutine BLANKS.

(Status $C000 indicates cursor is at end of screen). Checks if terminate
bit ATT/$17/4 has been set and if so jumps to REJSUP with A = 39.
Otherwise return to caller via jump to EXTSR.

INTERFACE

1. ENTRY METHOD

PPR is a disk resident program which is called via ENTPRG.

2. INPUT PARAMETERS

PPR command as entered by supervisor is stored in IRB buffer attached
to supervisor ATT. I = current ATT address.

3. EXIT METHOD

   Error exit by jump to REJSUP with appropriate error code in A. Normal
   exit after displaying or printing last line by jump to SV. When function
   is terminated by cancel key on CRT or MI, SX on TTY jump to REJSUP with
   A = 39. When function is terminated at end of screen on CRT by cancel
   key jump to SV.

4. OUTPUT

   One of the 4 header lines described in the detailed description of
   PROGRAM FUNCTION followed by as many lines as necessary containing the
   existing program numbers of the directory requested.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      ZERO ($22)
      ONEBIT ($23)
      TEN ($46)
      SST ($47)

   b. SST

      PRD (3)
      STEMP (20)
      ENTSR ($91)
      EXTSR ($92)
      SUPRW ($97)
      GETALP ($9B)
      GETCOM ($9C)
      SV ($B1)
      REJSUP ($B2)

   c. EST

      MAXOC (4)

   d. ATT

      TERMIN (23)
      AINT (27)
      SCA (36)

e. SCA

SLIST (SCA+1)
DIRTYP (SCA+10)
DIRIND (SCA+10+1)
CRTSTS (SCA+10+2)
DIRADD (SCA+10+3)
DIRLIM (SCA+10+4)
DIRIDX (SCA+10+5)
LINIX (SCA+10+6)

f. PRD

OWNC (4)
MAGTAP (7)

g. CONSTANTS

C ($43)
L ($4C)
O ($4F)
T ($54)

FW (3)
MOT (5)

2. PROGRAM DEPENDENCIES

PIO	SUPRW	For sending display line to output device and for WEOF motion request to clear CRT or advance page on output device.
	GETBUF	Get interrogate buffer.
SCANV	GETALP	Get next alpha character from input string.
	GETCOM	Get comma, EOR, or nonspace character from input string.
SUPER	REJSUP	Supervisor reject procedure.
	SV	Supervisor command processor.
ENEXSR	SETMOV	Enable disk resident program to move.
	EXTSR	Common exit to last entry in address stack.
	ENTSR	Common routine to save return address.
CONCAN		Checks for continue or cancel key at end of CRT.

3. DATA AREAS USED

ATT (SCA)
PRD

DOCUMENT CLASS   CYBERDATA IMS

PROGRAM NAME   PPR - Print User Program   Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$FA_{16} = 250_{10}$ words.

2. RELOCATION TYPE

Run anywhere, disk resident.

3. ENTRANCY CLASSIFICATION

Reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___PRD - Program Directory_____Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The PRD is a table saved on disk and called into core by VLSTRT. It contains
    an address for each disk resident routine, and a control block to indicate
    whether it is a disk address or a core address. When a disk resident routine
    is read from disk into core, ENTPRG updates its entry in this table. When
    a disk resident routine is loaded into the system by the supervisor command
    LPR, the corresponding entry in the PRD in core and on disk is updated to
    point to the new address of this routine.

2. DETAILED DESCRIPTION

    The address in the PRD are divided into sections according to the following
    type Programs: Miscellaneous, Key-functions, Interrogate, Supervisor,
    Mag-tape (including mag-tape own-codes, labels and code conversion) and
    Own codes.

    The first part of the PRD contains a pointer to the beginning of each
    section followed by a control block. Each bit in the control block
    specifies the type of address for the corresponding entry.

    If bit = 0   the entry contains a disk address. The disk address will
                   be zero while the program is being read into core by ENTPRG.

    If bit = 1   the program is in core, and the entry contains its core
                   address, note that in this case the disk address is saved in
                   the leader of the program.

    Each entry in the PRD is declared as an external (EXT) and its value is
    given at installation time. The entries which are not patched at
    installation time are given the value of $7FFF_{16}$.

INTERFACE

The fourth word of SST points to the first word of the PRD.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

2. RELOCATION TYPE

    VLSTRT relocatable.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___ PSS - Print System Status _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   PSS processes the supervisor function to print system status as follows:

   - % FREE MEMORY
   - % TIME CONTROLLED MEMORY
   - % OF DISK X USED, DISK (IN) ACTIVE, (DE) SELECTED
   - X ACTIVE JOBS IN SYSTEM.

2. DETAILED DESCRIPTION

   a. (PSS). Call GETCOM to scan for an EOR after the command.

      1) If no EOR is found, exit to REJSUP with A = 9 (invalid command format).

      2) Get starting address of FRP in core (SST + LL).

   b. (LOOP1). Check user count (word 2 in FRP).

      1) If area is used go to NXTAR.

      2) If area is not used, check time control bit (FRP word 1, bit 14).

      3) If time control bit not set, get length of area (word 1 of FRP), add to free area counter (FACTR).

      4) If time control bit set, get length of area and add to time control area counter (TCCTR).

      5) Go to NXTAR.

   c. (NXTAR). Advance to next area.

      1) Get length of area (word 1 of FRP), add to starting address of FRP buffer.

      2) Compare with FRP pointer which points to first free address.

      3) If there is another free address, go back to LOOP 1.

      4) If no other free address:

         - Get total available area (FRP upper limit - FRP lower limit); store in ATEMP.

         - Get free area left from pointer to upper limit of FRP; accumulate to FACTR.

      5) Go to CONVRT with A = Total available area + RACTR and ATEMP = total available area.

d. (CONVRT)

    1) If A = $\emptyset$, convert to ASCII $\emptyset$.

    2) If A = $\emptyset$: $\dfrac{A \times 100}{ATEMP}$; if result is $\emptyset$, convert

        to ASCII $\emptyset$. If result is not $\emptyset$, check if it is 100 or less.

    3) Return jump through (CONVRT).

e. 1) Plug result obatined in message "X % free memory".

    2) Return jump to SUPRW to output message.

f. Get area in time control.

    1) Return jump to CONVRT to get percentage of time control memory,
        with A = TCCTR.

    2) Plug result in message "X% time - controlled memory".

    3) Return jump to SUPRW to output message.

g. 1) Set TRTIDX (track reservation table index) to $\emptyset$.

    2) Calculate ATT limit address (ATTLIM)
        number of terminals X ATT length + ATT base address.

h. (NXTRT)

    1) Return jump to TRKSR with TRTIDX = $\emptyset$.

    2) Get TRT1 length (word $\emptyset$ of EST, bits 8-15). If $\emptyset$ go to OUT.

    3) Get first word of TRT1 to have number of tracks assigned
        to CONVRT with A = number of tracks assigned and ATEMP = length of
        TRT. Result is percentage of used tracks.

    4) Plug percentage in message TMSG
        "X % of disk Y used, DISK(IN)ACTIVE, (DE)SELECTED".

    5) Get disk number; if TRTIDX = 0, disk is 1 and plug it in message
        TMSG.

    6) Get address of first ATT. Go to ATTLUP.

i. (ATTLUP)

    1) Get current mode of ATT (word 2 of ATT, bit 8-11).

    2) If idle, check if it is last ATT by comparing to ATTLIM.

        ●   If last ATT, disk is inactive; get disk status (word 1D
             of EST, bits 4-7).
             Plug "in" and status deselected or selected in message.

        ●   If not last ATT, go to next ATT (SST + ATTLNG) and back to
             ATTLUP.

3) If not idle, get disk number and check if = to current disk. If yes, disk is active. If not, check if it is last ATT (plug parameter in message).

- Yes: disk inactive (plug "in" in message).

- No: Go to next ATT and back to ATTLUP to check mode.

4) Output message through SUPRW and go to OUT.

j. (OUT). Jump back to NXTRT + 1.

k. 1) Increase TRTIDX by 1. Check if = 3 (last TRT).

2) No: return jump to TRKSR (back to g. 2).
Yes: last TRT. Go to L.

l. Get legal job directory disk address.

- Call CLCDWA to calculate disk word address.

- Call REDDWA to read 1st word of LJD which contains number of active jobs.

- Convert number to ASCII and plug in message "X ACTIVE JOBS IN SYSTEM".

- Output message with call to SUPRW.

m. Exit to SV.

INTERFACE

1. ENTRY METHOD

PSS is a disk resident program which is called via ENTRG (enter a disk resident program).

2. INPUT PARAMETERS

PSS command entered by operator.

3. EXIT METHOD

Jump to (SV).

Jump to REJSUP when no EOR after command.

4. OUTPUT PARAMETERS OR CONDITIONS

System status printed.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___PSS - Print System Status_____Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        SST ($47)
        LPMASK (2)
        TEN ($46)
        ZERO ($22)
        ONEBIT ($23)
        GETCOM ($90)
        REDDWA ($A9)
        SV ($B1)
        CLCDWA ($90)
        REJSUP ($B2)
        SUPRW ($97)

    b. SST

        LL (6)
        UL (7)
        P (8)
        ATTB (10)
        ATT (11)

    c. EST

        TRT1 (11)
        LJDDA (8)

    d. ATT

        FACTR (50)
        TCCTR (51)
        TRTIDX (52)
        TRTAD (53)
        TLIM (54)
        TRKCTR (55)
        ATEMP (73)
        DLU (35)
        DSKST (29)

AA5641   PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___PSS - Print System Status_____ Level 6

2. PROGRAM DEPENDENCIES

SCANV - (GETCOM) get EOR after command
SUPER - (REJSUP) Reject supervisor
      (SV)   Conclude supervisors function request
PIO - (SUPRW) Read from/write to supervisor devices
PIO - (CLCDWA) Calculate disk word address
      (REDDWA) Read disk word address

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE

$111_{16} = 273_{10}$ words.

2. RELOCATION TYPE

Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

AA5641

PRINTED IN THE USA

DOCUMENT CLASS ___PST2___

PROGRAM NAME___Print Statistics Report Second Overlay___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   PST2 module is the second overlay of the PST command. It performs the printing
   of the contents of the accumulated records, transferred from the PST module,
   (the first overlay of PST command).

2. DETAILED DESCRIPTION

   STEP 1.

   PST2 receives control from PST, after accumulating the statistics records from
   the tape on the disk. It checks user count not to be greater than one (See
   3 under PHYSICAL CHARACTERISTICS), checks page count, line-count and output
   device, to be prepared for the printing. According to line-count and output-
   device it decides if the headers of the top of a page must be printed. If the
   answer is positive go to Step 2, otherwise to Step 3.

   STEP 2.

   Prints the headers after updating them.

   STEP 3.

   If a Daily report is requested the information of each day is transferred
   from RECBUF, (the buffer in which PST accumulates the records of each day),
   to the Interrogate buffer - from which the contents are printed. During the
   transfer the records are accumulated in 12 words, (from RECBUF + 18), and the
   contents are adjusted to the format of printing, and are printed. Control
   returns to PST at entry point START2 where next day Processing begins, except
   in the case of the last day of the statistics file, in this case the total
   line is printed passing the same stages as every daily record in PST2.

   If an Operator report is requested the record of each operator is transferred
   from the disk to RECBUF. From this point the execution is the same as in a
   Daily report, only, after the printing, instead of returning control to PST,
   PST2 retrieves the next operator's disk address and transfers its record to
   RECBUF, etc.

   If Jobs report is requested the record of each Job is transferred from the
   disk to RECBUF. The address of every Job's record is calculated before every
   transfer, because the Jobs are classified before transferring. From this
   point the execution is the same as in Operator report.

DOCUMENT CLASS ___PST2___

PROGRAM NAME___Print Statistics Report Second Overlay___

INTERFACE

1. ENTRY METHOD

    PST2 is a disk resident program which is called PST performing the following steps:

    ```
    EQU     ENTPRG ($9E)
    CLR     A
    LOQ     =N$0812
    RTJ-    (ENTPRG)
    ```

2. INPUT PARAMETERS OR CONDITIONS

    a. ATT communication area (46-63) will contain the following information:

SRMOD	- DOJ (BITS 0 AND 1):	00	- If Daily Report.
		01	- If Operators Report.
		10	- If Job Report.
	- EV (BITS 2 AND 3):	00	- If Entry + Verify Report.
		01	- If Verify Report.
		10	- If Entry Report.
	- PT (BITS 4 and 5):	00	- If print device is line printer.
		01	- If print device is TTY.

    DDMM1 (47) - Day of the start date and month of the start date.
    DDMM2 (48) - Day of the finish date and month of the finish date.
    LCOUNT (49) - Lines counter - on entry to PST2 LCOUNT = 0. If Jobs Report, Operator Report or the beginning of Daily Report. If Daily Report, (and not the beginning of the report), LCOUNT=No. of lines printed from the top of a page (including the headers).
    PCOUNT (50) - Page counter - zero if the beginning of the Report.
    NJOBS (51) - Number of jobs.
    DWA (55-62) - Disk Word Area - Tracks addresses of the disk word area.
    JDIRDA (63) - Jobs directory disk address (for Jobs Report).

    b. ATT Equates:

    JIDX (3)   - Job index.
    RIDX (4)   - Record index.
    JDIR (5)   - Jobs Directory Buffer - contains Job names and indices (4 words saved for every Job).
    INFUB (6)   - If a Daily report, and first word of INBUF is zero then TOTAL line is to be printed and control is transferred to PST.
    RECBUF (29) - If Daily report, RECBUF contains the accumulated record for each day.

    c. Accumulated records in tracks addressed in DWA for operators report and Jobs report.

DOCUMENT CLASS ___PST2_____

PROGRAM NAME_____Print Statistics Report Second Overlay_____

3. EXIT METHOD

There are 3 exits from PST2, all are to PST.

```
RTJ - (EXDFUN)
ENA   X
LDQ   #=N$3827
RTJ - (ENTPRG)
```

X = 1 when exit to PST during daily report.
X = 2 when exit is to completion exit of PST.
X = 3 when exit is to error exit of PST, after printing error message.

4. OUTPUT PARAMETER OR CONDITIONS

a. Error codes are transferred to the error program through the A register.

A = 39: FUNCTION TERMINATED - when SX is performed.
A = 69: FUNCTION BUSY - when user count is greater than 1.

b. Output layout of PST2:

1) Headers
    ## - page number

    1a) xxxx CYBERDATA STATISTICS REPORT (DDMM1 - DDMM2)

        xxxx = VER. if PST, $\begin{matrix}(D)\\(O)\\(J)\end{matrix}$, V

        xxxx = ENT. if PST, $\begin{matrix}(D)\\(O)\\(J)\end{matrix}$, E

        Blanks if both entry and verify are required: PST, $\begin{matrix}(D)\\(O)\\(J)\end{matrix}$

        DDMM1, DDMM2 - see a of INPUT PARAMETERS OR CONDITIONS.

    1b) (aaaa) KEYING NO. OF ERRONEOUS VERIFY   KEYS KEYS KEY
        (bbbb) TIME   RECORDS RECORDS   CORRECT HOURS RECORD STROKES   /
        aaaa = JOB
        bbbb = NAME, OP. NO, OR DATE

    1c) ---

2) Prints the content of the accumulated records, according to the header
    described in 1b).

340

DOCUMENT CLASS ____PST2_____

PROGRAM NAME_____Print Statistics Report Second Overlay_____

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. SST Equates

      SST ($47)
      STEMP ($20)
      TEMP1 (20)
      TEMP2 (21)
      TEMP3 (22)
      TEMP4 (23)
      REDSECC ($A7)
      WRTSEC ($A8)
      ENTPRG ($9E)
      EXDFUN ($9F)
      ENTSR ($91)
      EXTSR ($92)
      SUPRW ($97)

   b. ATT Equates

      JIDX (3)
      REDX (4)
      JDIR (5)
      INBUF (6) -(detailed description in b of INPUT PARAMETERS OR CONDITIONS)
      CNCL (23)
      AINT (27)
      OUTBUF (28)
      RECBUF (29)
      SCA (46)
      ATEMP (73)

   c. Communication Area Equates

      See b of INPUT PARAMETERS OR CONDITIONS.

   d. System Equates

      LPMASK ($2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)

341

DOCUMENT CLASS ____PST2____

PROGRAM NAME____Print Statistics Report Second Overlay____

    e. General Equates

      TOTAL (18) - RECBUF + TOTAL = The beginning of the buffer where the total contents of the statistics are accumulated.

      PAGLP (57) - No. of lines available in a page for line printer.

      PAGTTY (57) - No. of lines available in a page - for TTY.

      SPACE (9)   - No. of lines to space, when TTY, to get to the beginning of a new page.

      FW (3)   - FWRITE code on entry to PIO.

    f. Error Codes

    Described in 4 of INTERFACE.

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

    Entry points

    None.

    External list

    SETMOV - Entry in ENEXSR enables moving of a disk resident program.
    PRTMSG - Entry in SUPER, prints the message.
    BD - ARITH - converts binary number to decimal.
    DB - ARITH - converts decimal number to binary.
    MOVREC - Entry in PIO, moves records.

    Routines called via transfer vector

    REDSEC ($A7) - Entry in PIO.
                Reads sector addressing.

    WRTSEC ($A8) - Entry in PIO.
                Writes sector.

    ENTPRG ($9E) - Entry point to ENTPRG.

    EXDFUN ($9F) - Entry point in ENTPRG.
                Releases the program.

    ENTSR ($91) - Entry in ENEXSR.
                Saves the address of subroutines with I/O or relay in the exits stack A, Q and I are saved.

    EXTSR ($92) - Entry in EXEXSR.
                Exit subroutine used by subroutine which entered by ENTSR, entry by JMP and exit to last address in the stack.

    SUPRW ($97) - Entry in PIO.
                Called to read or write from or to suitable logic units.

342

DOCUMENT CLASS ___PST2_____

PROGRAM NAME_____Print Statistics Report Second Overlay_____

4. DATA AREAS AND SYSTEM TABLES USED

   a. Communication area

     This area is used to save the command parameters, counters and the tracks addresses of the disk work area (see a of INPUT PARAMETERS OR CONDITIONS).

   b. Accumulated record buffer - RECBUF

     This buffer is used to store the current accumulated record and the totals counters.

   c. INBUF buffer

     This buffer is used by the PST program. The PST2 program checks only the first word of this buffer.

   d. Jobs directory buffer

     8 Sectors buffer for the jobs directory.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $1084 = 43C_{16}$ words for the program.

      653   words for the buffers.

2. RELOCATION TYPE

   Run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrancy locked.

343

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ PTS - Print Terminal Status _____ Level 06 _____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

The PTS prints the current status of each terminal requested. Information displayed includes the display and keyboard type, the mode, job name, batch number, and lock indicator.

2. DETAILED DESCRIPTION

PTS calls SETMOV to enable the program to move, and then performs syntax checks on the parameters entered. If an error is detected (commas missing, terminal numbers are valid or greater than the maximum specified) exit is to REJSUP with an appropriate error message. Next the header is moved to the interrogate buffer, and is printed via SUPRW.

A larger interrogate buffer (18 words) is requested via GETBUF, and the attributes of the first terminal requested are stored in it. Each line is printed via SUPRW, after which a check is made for the cancel key, and if it was entered (or SX from the TTY) the program exits to REJSUP. If end of screen condition is reached, CONCAN is invoked to wait for CANCEL or CONT.

When the information for the last terminal has been printed, the program exits to SV.

INTERFACE

1. ENTRY METHOD

PTS is a supervisor function loaded by ENTPRG. Execution begins at its only entry point PTS.

2. INPUT PARAMETERS OR CONDITIONS

None.

3. EXIT METHOD

a. Normal exit to SV.

b. Error exit to REJSUP with appropriate error message.

4. OUTPUT PARAMETERS OR CONDITIONS

See Cyberdata Reference for sample printout.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___PTS - Print Terminal Status_____Level 06____

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

TEN	($46)
ZERO	($22)
NZERO	($12)
ONEBIT	($23)
LP MASK	($ 2)
SST	($47)

    b. SST

ATTLING	(19)
CATT	(11)
ATTB	(10)
SLOCK	(1)
S TEMP	(20)
SUPRW	($97)
REJSUP	($B2)
SV	($B1)
GETBIN	($99)
GETCOM	($9c)
GETSTR	($A2)
CAM	($AE)
GETALP	($9B)

    c. ATT

CMODE	(2)
JIDX	(3)
BATCHN	(35)
BMOD	(15)
TERMIN	(23)
CNCMT	(46)

TERN   (47)

ATTAD   (48)

TEMP   (49)

JNM1   (50)

AINT   (27)

TRMNT   (23)

OUTCU   (37)

d. INTERROGATE BUFFER

TER   (5)

DISPTY   (6)

KB   (8)

MODE   (11)

JOBN   (15)

BATCH   (19)

LOCK   (21)

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

SETMOV

GETBUF

CONCAN

GETLJD

4. AREAS AND SYSTEM TABLES USED

SST

ATT

LJD

IRB

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___PTS - Print Terminal Status_____Level 06_____

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$132_{16} = 306_{10}$ Program length.

22_{10} Interrogate buffer.

2. RELOCATION TYPE

Disk resident, relocatable.

3. ENTRANCY CLASSIFICATION

Reentrant.

DOCUMENT CLASS ___RDMODS_____

PROGRAM NAME_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    RDMODS displays the page that contains the current field for 8 function keys. Six entry points service the function keys as follows:

    CBSRRR: BSR/REL/READ

    CBORKY: BOR

    CBSF: BSF

    CSKPR: SKIP

    CBSCR: BSC

    CSPACE: SPACE

2. DETAILED DESCRIPTION

    a. CBSRRR

    - Call ENDSR to save return address in Exits stack.

    - Call FINDFD to locate page and line of current field.

    - Go to step c.3. to display page with current field.

    b. CBORKY

    - Call ENDSR to save return address in Exits stack.

    - Call GETPL to locate page and line for last field prior to BOK key.

    - If field is not on first page, go to step c.4. to display first page.

    - Otherwise, call FINDFD to locate page and line of fields and go to step c.5.

    c. CBSF

    1. (CBSF). Call ENDSR to save return address in Exits stack.

    - Call FINDFD to locate page and line of current field.

    - If last page of record, go to step c.3. to display current page.

    2. (CBSF11). If field is not at top of a page, go to step 5.

    3. (PAGE). Save current field number in ATT/RQ.

    4. (PAGE1). Call CPAGE to display page that contains field indicated in ATT/RQ.

    - Exit to caller via EXDSR.

276

    5. (CURSOR). Advance screen address of field by 3 to allow for error flag, and save it in TOPT/FLDST.

    6. (CRSRl). Save screen address of field in TOPT/CURSRD.

- Call CCOL to display column number of current field.

- Call CFNDTY to display field number and data type.

- Call CONCSR to convert cursor position to line number and position in line.

- Call CDATA to move cursor to position where field data is to be displayed.

- Exit to caller via EXDSR.

d. CSKPR

    1. (CSKPR). Call ENDSR to save return address in Exits stack.

    2. (CSKPO). Call FINDFD to locate page and line of current field.

- Go to step c.2. to display page with current field.

e. CBSCR

- Call ENDSR to save return address in Exits stack.

- Back up current screen position by 1, and go to step c.6. to adjust display.

f. CSPACE

- Call ENDSR to save return address in Exits stack.

- If ATT/FCNT/0-7 = 0 (Beginning of field), go to step d.2. to display page with current field.

- If not, increment current screen position by 1, and go to step c.6. to adjust display.

INTERFACE

1. ENTRY METHOD

    Each routine is called by RTJ.

2. INPUT PARAMETERS OR CONDITIONS

    (I) = Base address of Active Terminal Table (ATT).

3. EXIT METHOD

    Return to location immediately following RTJ instruction.

4. OUTPUT PARAMETERS OR CONDITIONS

    (I) = Base address of ATT

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ZERO ($22)
        CDATA ($8D)
        ENDSR ($8E)
        EXDSR ($8F)

    b. ATT

        FCNT (3)
        FLDN (14)
        TOPT (33)
        RQ (65)

    c. TOPT

        CURSRD (3)
        FLDST (15)

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

CMISC1	{ ENDSR	Save return address in Exits stack
	ENDSR / EXDSR	Return to last address in Exits stack.
VL2CRT	{ CDATA	Display designated message
	CONCSR	Convert cursor position to line number and position within line
HDRNOS	{ CCOL	Display column number of current field
	CFNDTY	Display field number and data type
GETPL		Get parameters for current field.
CPAGE		Display page with current field.

4. DATA AREAS USED

    ATT   Active Terminal Table
    TOPT   Terminal Output Table

DOCUMENT CLASS __RDMODS__

PROGRAM NAME_____

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $4D_{16} = 77_{10}$ words

2. RELOCATION TYPE

    Core resident, loader relocatable.

3. ENTRANCY CLASSIFICATION

    Passive reentrant

279

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME_____ REDKEY _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   REDKEY performs the switching from basic mode to READ mode, upon depressing
   the READ key. The necessary information from basic mode is kept so return
   is possible. Partially entered records are redisplayed and the terminal
   is placed in the READ mode.

2. DETAILED DESCRIPTION

   a. Call SETMOV to allow program movement in core.

   b. If the pseudo-entry bit is on or the terminal is already in READ mode
      an "ILLEGAL FUNCTION KEY" error message is generated. Same error
      occurs in Entry if the character count in the current field is not 0,
      or if no records were entered in current batch. For partially entered
      records, "BASENT" writes it on disk, and updates all the needed pointers.

   c. When basic mode is verify: mismatch character is cleared, the IDK bit is
      set to 0; if the REVALIDATE bit (SW2B13) is not set, "BOR" continues
      processing.

   d. If counters are not required for current field, "BOR" continues
      processing. Otherwise, RCNT is set to point to the end of current
      field, character counter in current field is set to full length of
      that field. If counters are used for that field, they are updated.

   e. The record count and field count are set to their previous values.

   f. "BOR": If SW1B13 is set, the record is rewritten on disk. If
      SW2B11 is not set, "SETATT" continues processing. Otherwise, action
      is like entering READ mode at end of record; the most significant bits
      of next record's address are added to the m.s.b. of current disk
      address. The current disk address l.s.b. are moved to word 4 of TRB
      and SETATT continues processing.

   g. "BASENT": Current record is written on disk.
      If there is not an inserted record after the last one, the available
      track address is decremented by the record size.

   h. The next available disk address is updated to point after the partial
      entered record.

i. "SETATT": The current disk address is set as the basic mode disk address.

Current mode is set to READ.

The record counter is saved in basic mode counter.

The number of records in READ mode is set to the actual number of records in batch.

The format number is saved in basic mode format number.

Bits 15-13 and 5 of switch 2 are cleared, as well as bits 9, 6, 5, and 2 of switch 1.

After displaying the READ mode header, REDKEY exits through CLRSR.

INTERFACE

1. ENTRY METHOD

Entered by MPC upon detection of that key-function.

2. PARAMETERS

I = ATT base address.

3. EXIT METHOD

Either to ERROR or via CLRSR.

4. OUTPUT PARAMETERS

Last record written on disk, ATT pointers set so as to allow return to basic mode.

ASSEMBLY CHARACTERISTICS

1. EQUATES

a. LOCORE

LPMASK (2)
ZERO ($22)
ZROBIT ($33)
NZERO ($12)
ONEBIT ($23)
SST ($47)

    b.   SST

         CFFA (12)

    c.   ATT

         ATEMP (73)
         ATRB (6)
         CURDA (9)
         FCNT (3)
         RCNT (4)
         SW1 (19)
         RCDSIZ (69)
         BMODE (15)
         FMTNO (13)
         AVLTK (8)
         NDA (11)
         BMDA (17)
         CMODE (2)
         RCD (32)
         SW2 (20)
         BRCNT (15)
         RCDRD (31)
         BFMTNO (25)

    d.   TRANSFER VECTOR

         ERROR ($96)
         WRITEC ($A3)
         REWRIT ($A4)
         DISP ($A0)

    e.   EXTERNALS

         SETMOV
         DREADK
         COUNT
         FMTPAR

2.   DATA AREAS USED

     TRB
     SST
     ATT

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____REDKEY_____ Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $82_{16} = 130_{10}$ words.

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME_____ REDOC - Read A Document   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    REDOC reads a format entry for the current document. There are two entries
    to REDOC: REDOC and REDGRP. A call to REDOC causes the subroutine to locate
    the document, using the Legal Document Directory and Legal Document Table,
    and reads the first format for the document.

    A call to REDGRP causes the subroutine to read the next format for the
    document.

2. DETAILED DESCRIPTION

    a. REDOC entry

        1. Compute address of pointer for current document in the Legal
           Document Directory (LDD).

        2. Read pointer word from mass storage.

        3. If pointer word = 0, document is nonactive. Set ATT/SW1/3 and
           ATT/SW1/10 to insure that Document or Format Select will be keyed next.
           Display error code D3 (Invalid Doc. No.) and exit via error.

        4. If pointer word ≠ 0, set ATT/SW1/14 (first record in document flag),
           display document number and continue at REDGRP entry.

    b. REDGRP entry.

        1. Get address of current group in document.

        2. Read 2-word group entry from mass storage.

        3. Put number of records in current group in ATT entry CRCD in
           complement form, except for value of zero, which is not complemented.

        4. If this is the last group in the document, set ATT/ELDT/15 (last group).

        5. Put Format number in ATT word DOCFMT.

        6. Call GFCORE to locate the required format, and read it from disk,
           if it is not already in core.

        7. If the format number is invalid, set ATT/SW1/3 to insure that another
           format will be selected either by FMSEL, DOC or by FMT.

        8. If the format number is valid, reset ATT/SW1/3, set ATT/SW1/15 (first
           record in format flag).

        9. Set data type and the display for the first field to be entered.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___REDOC - Read A Document_____Level 6

INTERFACE

1. ENTRY METHOD

   REDOC is a disk resident program which is called via ENTPRG (enter a
   disk resident program), used as a second overlay of DOC and is called also
   by EORENT.

2. INPUT PARAMETERS OR CONDITIONS

   (1) = Address of the Active Terminal Table (ATT)

   DOCNO in the ATT contains the required document number. When REDGRP
   is entered ELDT in the ATT contains the displacement in the DOC table
   for current format.

3. EXIT METHOD

   Exit is via CLRSR (Clear exit subroutine), which exits to dispatcher.

4. OUTPUT PARAMETERS OR CONDITIONS

   (1) = Address of ATT

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)
      SST ($47)
      CLCDWA ($90)
      CLRSR ($93)
      ERROR ($96)
      GFCORE ($9D)
      DISP ($A0)
      REDWA ($A9)
      WRTATT($AC)
      RELFOR ($AD)

   b. SST

      EXTSST (0)

   c. EXTSST

      LDDDA (9)

    d. ATT

        PFBA (7)
        FMTNO (13)
        ELDT (16)
        DOCNO (17)
        SW1 (19)
        DOCFMT (23)
        CRCD (24)
        RCD (32)
        ATEMP (73)

    e. ERROR CODES

        CD3 ($F)
        CF1 ($14)

    f. EXTERNALS

DFMT	Display format number (entry point in DISPLY)
DDOC	Display Document number (entry point in DISPLY)
CCRFLD	Clear field (entry point in DISPLY)
DBOF	Display beginning of field (entry point in DISPLY)
SETMOV	Entry point in ENEXSR
FIRSTF	Entry point in EORENT

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

ENEXSR	SETMOV	Enable move of disk resident program
PIO	CLCDWA	Calculate Disk Word Address
	REDWA	Read Disk Word Address
DISPLY		Display a message
	RELFOR	Release Format
GSE	ERROR	Processor
GFCORE		Get Format into core
	WRTATT	Write Active Terminal Table (ATT) check point for power fail recovery)
	CLRSR	Clear subroutine stack and exit to dispatcher
EDRENT	FIRSTF	Locate first data entry field in format

4. DATA AREAS USED

SST	(System Status Table)
EST	(Extend System Table)
FBA	(Format Base Address Table)
FMT	(Format Table)
ATT	(Active Terminal Table)

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___REDOC - Read A Document___   Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $76_{16} = 118_{10}$ words

2. RELOCATION TYPE

    REDOC is a Disk Resident program. It is written in run-anywhere form, and movable.

3. ENTRANCY CLASSIFICATION

    Reentrant code.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA_____

PROGRAM NAME____REG - Regenerate Display_____ Level 06____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   The program processes the interrogate function REG which regenerates the display.

2. DETAILED DESCRIPTION

   The program checks validity of command. In case of incorrect syntax, exit is made to ERROR (ILLEGAL REQUEST). Otherwise, DISPLAY is invoked to regenerate the screen.

INTERFACE

1. ENTRY METHOD AND PARAMETERS

   REG is a disk-resident program invoked by INTRGT (via ENTPRG).

2. EXIT METHOD

   a. To Error with Q = $12

   b. Normal Exit is to CLRSR.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   SST   ($47)

   STEMP   (20)

   GETALP ($98)

   DISP   ($A0)

   ERROR   ($96)

   CLRSR   ($93)

   CE2   ($12)

2. EXTERNALS

   REGEN

   SETDIS

   MACROS USED

   None

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___REG - Regenerate Display_____Level 06___

3. PROGRAM DEPENDENCIES

GETALP

ERROR

DISP

CLRSR

4. AREAS AND SYSTEMS TABLES

SST (SCANV uses SST + STEMP for current pointer)

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

11_{16} = 17 words of program.

2. RELOCATION TYPE

Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

RELKEY performs the entry/verify release key operation by exiting
to RELIT. In Read mode, the next record is read, with ATT pointers
set accordingly.

2. DETAILED DESCRIPTION

SETMOV is called to allow the module to move. The current mode and
status bits are checked: if the mode is Pseudo-Verify, or the revalidate
bit (ATT/SW2/13) is set, exit to ERROR.

If the mode is Pseudo-Entry and an insert is not begin done or insert is
being done in basic mode verify, exit to ERROR. When exiting to ERROR,
Q = 4 (A4-illegal function). The mode is checked to be read. If it is,
control goes to READRL. Otherwise ATT/SW1/9 is cleared ("skip must be
next"), ATT/SW2/5 is set (REGEN if error occurs), the program is released
with a call to EXDFUN, the IDK bit is cleared, and the program exits to
RELIT.

READRL - ATT/ATEMP is set to zero. If a rewrite is required, REWRITE
is called. End of data is tested: if the basic mode verify and the
current record is the last in the batch, exit to ERROR. If the basic
mode is Entry and ATT/RCD-ATT/RCDRD = 0 (the difference is saved in
ATT/ATEMP), exit to ERROR. In both cases, on exit Q = 7 (B2-end of
data). In the case of the basic mode entry, if ATT/ATEMP = 1
(difference between the basic record count and the record count in
read), then the basic column number (ATT/BRCNT/0-12) is tested. If it
is zero, then there is no next record. The current disk address (ATT/
CURDA, CURDA + 1) is saved in the TRB as the previous disk address,
ATT/RCDRD is incremented and control goes to RCDST.

In the normal read mode case, that is, not on the last record of the
batch, ATT/RCDRD is incremented, the next record is read with a call
to READN, and the format of the new record is extracted from the TRB
header.

If the format is the same as the format of the previous record, and
ATT/ATEMO ≠ 1, jump to DISPIT. If ATT/ATEMP = 1 and ATT/BRCNT is
greater than ATT/RCNT, jump to DISPIT. Otherwise, jump to RCDST. If
the formats were different, the new format is loaded with a call to
GFCORE.

RCDST - ATT/FCNT. RCNT are cleared. ATT/RFA and SST/CFFA are initialized
to the first field in the format. STTYPE is called to initialize the
field data type.

DISPIT - DISP is called with DBSREL as a parameter (display backspace
record and release in Read mode). The program exits to CLRSR.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___RELKEY - Perform Release Function Key___ Level 6

INTERFACE

1. ENTRY METHOD

    RELKEY is a key function processor loaded by ENTRPG when requested by MPC. Execution begins at RELKEY.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT address.

3. EXIT METHOD

    Jump to ERROR (in GSE) with Q = 4 or 7 when the key is illegal.
    Jump to RELIT (SKPDUP) for normal entry/verify processing.
    Jump to CLRSR (ENEXSR) at the end of normal Read mode processing.

4. OUTPUT PARAMETERS OR CONDITIONS

    In Read mode, pointers have been advanced to the next record.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK ($2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)

    b. SST

        FFA (12)
        CLRSR ($93)
        ERROR ($96)
        GFCORE ($9D)
        EXDFUN ($9F)
        DISP ($A0)
        REWRIT ($A4)
        READN ($A6)

c. ATT

MODE (2)
FCNT (3)
RCNT (4)
ATRB (6)
PFBA (7)
CURDA (9)
FMTNO (13)
RFA (14)
BMOD (15)
BRCNT (15)
SW1 (19)
SW2 (20)
RCDRD (31)
RCD (32)
ATEMP (73)

2. PROGRAM DEPENDENCIES

ERROR (GSE) - exit taken when function illegal
REWRIT (PIO) - reqrites current record
READN (PIO) - reads in next record of the batch
GFCORE - link up this station to a format
EXDFUN (ENTPRG) - release the program
CLRSR (ENEXSR) - common end-of-processing exit
DISP (DISPLY) - central display call
DBSREL (DISPLY) - select Read mode BSR or REL display function
RELIT (SKPDUP) - exit to perform E/V REL function
STTYPE (EMM) - extract data type from format and store in ATT
SETMOV (ENEXSR) - allows program to move

3. DATA AREAS USED

ATT
TRB

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$6D_{16} = 109_{10}$ words.

2. RELOCATION TYPE

Run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____REPKEY_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

REPKEY is disk resident routine to process the REP FLD key. REPKEY is
illegal at end of data -EID- in the following modes: Verify, Pseudo
Verify, and Read, and COR must be next after pressing DLT key in Verify
or Read mode (not during insert).

2. DETAILED DESCRIPTION

REPKEY first clears the ERROR OVERRIDE FLAG (SW2-B12), SKIP KEY MUST
BE NEXT FLAG (SW1-B9), and IDK bit (ATT 2/14).

a. In ENTRY and Pseudo Entry modes:

● In Entry and Insert modes, at the beginning of record (RCNT=0),
no processing is performed - IDK bit is set and the routine exits
via RTJ - (CLRSR).

● Otherwise:

1) Record counter is set back to the beginning of current field.

2) If the following conditions:

a) Not at beginning of field.

b) Not in Entry mode.

c) It is not the first record in the format.

d) None of the records auto-dup fields was changed in the
Entry pass.

are all satisfied, then the DUPKEY bit (SW2-B4) is set.

3) If the REP FLD key was pressed in the first field of the
record (the record counter is zero at this point) then:

a) Set IDK and FIRST CHAR. OF RECORD NEXT (SW1-B2) bits.

b) If DUPKEY bit (SW2-B4) is set and it is not first record
in batch, read previous record from the disk and clear
DUPKEY bit.

4) Current data display is cleared via DISPLY-CLRFLD.

5) Set field counter to point to the beginning of field (store
zero in FCNT).

6) Disply column number via disply-DCOL.

7) Exit via CLRSR.

AA5641   PRINTED IN THE USA

b) In READ, Verify, and Pseudo Verify modes:

1) If the end of the data (EOD) is reached in Verify, Pseudo Verify, or Read modes, or when COR must be next (after DLT key was pressed in Verify or Read mode, not during Insert), the routine exits via ERROR with Q=4 - invalid function key. The check for EOD is mode via the subroutine TSTEOD.

2) Clear COR (B11) and CONT (B5) bits in SW1.

3) Set rewrite bit (SW1-B13) because a correction is to be made.

4) Clear IDK and P.V. (Pseudo Verify) bits in ATT,2.

5) Set P.E. (Pseudo Entry) bit in ATT,2.

6) If Basic mode is Verify, MISMATCH CHAR. (CATT,5) is cleared.

7) Update the counters via the subroutine CHKCNT.

8) Check flags and clear, if necessary, ERR bit in the Reader via the subroutine FLAGS.

9) Continue as in Entry mode (steps 4) - 7)).

2.1 DETAILED DESCRIPTION OF SUBROUTINES

a. TSTEOD

Subroutine that checks for End of Data. If return from the subroutine is with A=0, then EOD was reached.

1) In basic mode Entry:

If the total number of records in batch (RCD) is equal to the number of records in batch in Read mode (RCDRD) and the char. count in current record (RCNT) is equal to char. count in Basic mode (BRCNT), then return from the subroutine is with A=0, otherwise return is with A7=0.

b. CHKCNT

Subroutine that reverse the previous effect on the counters of current field.

1) The subroutine does a return jump to FMT PAR with Q=11. If, upon return, A is negative, there are no counters and return is made from the subroutine; otherwise A contains the value of the counters which is saved in ATEMP+1 in ATT.

2) Save RCNT in ATEMP + 2 and FCNT in ATEMP + 4 in ATT.

3) RCNT and FCNT are updated to point to the end of field.

4) The sign of the counters is <u>reversed</u>.

5) The subroutine does a return jump to COUNT. The values of the counters are added (or subtracted) if they were subtracted (or added) before.

6) The value of RCNT and FCNT are restored from ATEMP + 2 and ATEMP + 4.

c. FLAGS

This subroutine checks for a flag on the current field. If found, then it checks the whole record (in Entry and Insert made checks till current field in record). If no other flag is found, then the ERROR bit is found, then the ERROR bit in the TRB header is cleared, and the number of erroneous records is decremented.

1) The value of RCNT is saved in ATEMP+1 in ATT.

2) Check for a flag on the current field:

Field count (FCNT) is subtracted from record counter (RCNT) so that RCNT points to the first character fo the field. The character is read via GBYTE. If it is not flagged, return from the subroutine.

3) Clear the flag from current field in the TRB:

Clear the flag in CURCH in ATT and store it via SBYTE in TRB.

4) Initializations:

● Record count points to the beginning of record (store zero in RCNT).

● STEMP in SST gets the address of the beginning of format.

5) Read for character of current field via GBYTE. If it is flagged, return from subroutine, because there is still at least one erroneous field, and ERROR bit is the reader must remain set.

6) RCNT points to the beginning of the next field (max number of characters in current field is added to RCNT).

7) Check if the loop is finished:

● In Entry and Insert modes: reached from BSC or BSF. These programs pass to REPKEY the difference between the record and field counters in bits 0-11 of ATEMP, so that only fields from the beginning of record till current field will be checked for error flags.

If ATEMP is zero (at first field of the format) <u>or</u> the difference between ATEMP and RCNT is zero, the check is complete, and no flagged field was found, so jump to CLRERR (execute step 10). If not, continue the check (go to (*)).

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __REPKEY__   Level 6

- ● If basic mode is Entry: check if at the end of data via the subroutine TSTEOD. If so, jump to CLRERR (execute step 10).

(∗) If it is the last field in the format jump to CLLERR (execute step 10).

8) STEMP in SST points to the current address in the format (number of words in current field is added to STEMP).

9) Continue the check (execute again from step 5).

10) CLRERR - There is no flagged field in the record. ERROR bit in the TRB header is cleared. If the field replaced in pure Read mode, before the basic position, and it was the only erroneous field in the record, decrement the number of erroneous records (RCDER in ATT).

11) Restore RCNT from ATEMP+1 and return from the subroutine.

INTERFACE

1. ENTRY METHOD

REPKEY has five entry points:

REPKEY - processing of REP FLD key.

REP2   - used by CORKEY for checking counters.

REP3   - used by EMM to clear the error flag of the current field, and set rewrite bit (SW1-B13).

REP4   - used by BSE to check counters and flags. Field are checked from the beginning of the record up to the current field.

REP5 - used by BSF in the way as BSC.

REPDEY is loaded via ENTPRG when the operator hits the REP FLD key. Execution begins at the first entry point REPDEY.

2. INPUT PARAMETERS OR CONDITIONS

ATEMP contains the difference between the record count and the field count if REPKEY is entered from BSC or BSF.

3. EXIT METHOD

If REP FLD was depressed exit is either to CLRSR or to ERROR. Otherwise return is to the appropriate entry point of the calling routine, after releasing REPKEY via EXDFUN.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        EXTSR ($92)
        CLRSR ($93)
        GBYTE ($94)
        SBYTE ($95)
        ERROR ($96)
        ENTPRG ($9E)
        EXDFUN ($9F)
        DISPLY ($A0)
        READP ($A5)

    b. ATT

        MODE (2)
        CURCH (2)
        FCNT (3)
        RCNT (4)
        MISCH (5)
        ATRB (6)
        PFBA (7)
        CURDA (9)
        BMOD (15)
        BRCNT (15)
        SW1 (19)
        SW2 (20)
        RCDER (30)
        RCDRD (31)
        RCD (32)
        ATEMP (73) ·

    c. SST

        STEMP (20)
        CFFA (12)

2. PROGRAM DEPENDENCIES

    DISPLY - CLRFLD - Display cleared field.
            - DCOL   - Display column no.
    FMTPAR ·   - Check in the format parameters if counters are specified.
    COUNT   - Update the counters.
    READP   - Read previous record from disk.
    GBYTE   - Read a character from the TRB.
    SBYTE   - Store a character in the TRB.
    ENTPRG   - Enter a disk resident program.
    EXTSR   - Exit subroutine.
    EXDFUN   - Exit disk function.
    CLRSR   - Clear exit subroutine.
    ERROR   - Display terminal error.

3. DATA AREAS USED

    a. ATT

        words 2   - B12, B13, B14
        words 3   - FCNT B0-B7
        words 4   - RCNT
        words 5   - MISMATCH CHAR.
        words 19   - SW1: B2, B5, B9, B11, B13
        words 20   - SW2: B4, B12
        words 30   - RCDER
        words 36,63 - (8 words) counters
        words 73   - ATEMP
        words 74   - ATEMP + 1
        words 75   - ATEMP + 2
        words 77   - ATEMP + 4

    b. SST

        word 20   - STEMP

    c. TRB

        word 1   - B15 (error bit in the header)

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $124_{16} = 294_{10}$

2. RELOCATION TYPE

    Relocatable.

3. ENTRANCY CLASSIFICATION

    Reentrant

NOTE: The program is not movable during execution.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___REPKEY_____ Level 6

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME____RESET - Routine to Perform Reset Key Function_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    RESET clears an error lock condition and the error code display. For
    certain error conditions, status bits are set defining what the following
    sequence of keys must be. If there was no error lock the key has no
    affect on the system.

2. DETAILED DESCRIPTION

    SETMOV is called to allow the program to move, the error code (in
    ATT/ECODE) is cleared and ATT/ECODE is saved in ATT/ATEMP. The IDK bit
    is cleared; if ATT/SW1/8 is set, the display is regenerated. The error
    code on the display is cleared. If error lock (ATT/0/11) is not set,
    jump to EXIT. Otherwise, clear the error lock.

    The error code is recovered from ATT/ATEMP. If the code is less than
    8 (not a validation error) jump to EXIT. If the code is for a validation
    error, set ATT/SW2/12 (error override allowed). For any higher error
    codes, a check is made for ATT/SW1/3 being set (DOC, FMT ADV, or FMT SEL
    must be next). If the bit is set DISPLY is called to clear the current
    field and to display beginning of field.

    EXIT - ATT/SW1/2, 7, and 8 are cleared (regen, reset and beginning of
    record bits).

    Set the IDK bit under the following conditions:

    a. Mode is Idle, or

    b. Mode is Verify or Pseudo-Verify and ATT/MISCH $\neq$ 0 or ATT/SW2/11
       is set, or

    c. RCNT = 0 and mode is either Entry or an insert is being performed
       (set ATT/SW1/2 as well), or

    d. Any one of ATT/SW1/0, 2, 5, 9, 11 is set, or

    e. ATT/SW1/12 is set.

    The program exits to CLRSR.

INTERFACE

1. ENTRY METHOD

    RESET is a key function program loaded by ENTPRG when requested by MPC.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME____RESET - Routine to Perform Reset Key Function_____Level 6

3. EXIT METHOD

    Jump to CLRSR. The error code and error lock have been cleared.

4. OUTPUT PARAMETERS

    I = current ATT base address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ONEBIT ($23)
        ZROBIT ($33)

    b. SST

        CLRSR ($93)
        DISPLY ($A0)

    c. ATT

        MODE (2)
        RCNT (4)
        MISCH (5)
        SW1 (19)
        SW2 (20)
        ECODE (34)
        ATEMP (73)

2. PROGRAM DEPENDENCIES

    CLRSR (ENEXSR) - Common completion return.
    DISPLY - Central display call.
    DBOF (DISPLY) - Displays beginning-of-field condition.
    REGEN (DISPLY) - Regenerates display.
    DERR (DISPLY) - Clears the error code.
    CLRFLD (DISPLY) - Clears the current field.
    SETMOV (ENEXSR) - Allows program to move.

3. DATA AREAS USED

    ATT - Active terminal table.

AA5641

PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME__ RESET - Routine to Perform Reset Key Function __ Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$$5E_{16} = 94_{10}$$

AA5641

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME ___RSQ - Resequencing Program___   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    RSQ, on overlay for the WBT supervisor command, is called once for
    each data record, the record is resequenced, overpunch and fill
    character are performed, as is the variable recording function. The
    record is transferred from the WBT input buffer to the Terminal Record
    Buffer (TRB) one field at a time, either in the entered order or when
    resequencing is required in resequencing field number order.

    The subroutine FIELD performs the actual operations described above and
    is called once for each field.

2. DETAILED DESCRIPTION

    SETMOV is called to allow the program to move. The format of the
    current record is checked to be the same as the current format. If
    not, GFCORE is called to load in the format on the record. The EOD1
    buffer format table pointer (first word following the EOD1 image) is
    used to load the latest entry in the format table. If the entry is
    negative, store zero in ATT/ATEMP + 3 (format is already flagged). If
    the entry is positive and equal to the current format, save the
    address of the current entry in ATT/ATEMP + 3 (first call to RSQ with
    this format). Otherwise (current format not equal to last new format),
    store zero in ATT/ATEMP + 3.

    The number of fields in the format are counted and saved in ATT/NFLDS.
    If variable recording was requested on the mag tape format (MTF),
    subroutine IFON is called; then the TRB is checked to see if it is
    long enough: maximum record length is the record length in the format
    plus one character per field. If the current TRB is shorter than this,
    the TRB is released and CAM is called for a new buffer. If the call
    to CAM fails 100 times, RSQ prints "MEMORY FULL" and exits to MTFMON
    entry point 4 (abort entry). If the CAM request succeeds, the TRB
    backwards pointer is set up. The 5-word header for the record being
    processed is moved from the input buffer to the TRB. ATT/RCNT is set
    to zero. The first field on the format is checked for a resequencing
    field number. If there isn't one control goes to NORSQ. Otherwise,
    subroutine IFON is called and the record is processed, one field at a
    time, in resequencing field number order: SST/STEMP + 3 is set to 1
    (for field 1). The format is scanned from the beginning, looking for
    a field with the same resequencing field number as in SST/STEMP + 3.
    ATT/CHRPTR is increased by the length of a field if that field is
    passed over, so ATT/CHRPTR always contains the character position
    within the record of the current field. Q is set to the address of the
    current format field description. If the last field in the format is
    reached without a match, the system will hang, as there is an error in
    the format string.

When the resequencing field number matches SST/STEMP + 3, subroutine
FIELD is called. SST/STEMP + 3 is compared with ATT/NFLDS. If they are
equal, the record has been completely processed and control goes to
EVCHK. Otherwise, SST/STEMP + 3 is incremented, and the search begins
again at the start of the format.

- NORSQ - When there is no resequencing, ATT/CHRPTR is initialized to zero.
  Q is set to the address of the current format field description.
  FIELD is called. If this was not the last field, ATT/CHRPTR is
  advanced to the next field, Q is advanced to the next field
  description and the next field is processed.

- EVCHK - If the final record legnth (ATT/RCNT) is odd, the last character
  in the record is made a binary zero and ATT/RCNT is incremented.
  ((ATT/RCNT) + 1)/2 is now transferred to the TRB record header
  as the record length in words. The IDK bit (ATT/CURCH/14), which
  had been cleared by FIELDS, is restored. The program is released
  by calling EXDFUN and control is returned to the caller (MTFMON
  entry 1) by calling ENTPRG with Q containing the value in ATT/
  MTCA + 1.

a. Subroutine FIELD

FIELD is called once for each field in a record. On entry, Q = address
of the format field description and ATT/CHRPTR = column position of the
first character in the field. ATT/CHRPTR is saved as BEG. BEG plus the
field length + 1 from the format is saved as END.

BEG and END thus indicate the first and last character positions of the
field, while RCNT is used as the next available position in the TRB.
If overpunch is specified in the MTF and the field is signed numeric,
overpunch is performed: first IFON is called, then the overpunch option
in the MTF is examined: if minus-only overpunch is specified, control
goes to MINOVP. Otherwise, the last character in the field is checked
to be a minus. If it is control goes to L3. If it is a plus, ZERCOD is
set to BCDZEP or ASCZEP, depending on whether or not BCD is the specified
recording code. ADDPAR is set equal to PLSADD and control goes to L2.

MINOVP - If the last character in the field is minus, go to L3, otherwise,
decrement END (suppressing the plus sign) and exit to VR.

L3: set ZROCOD to BCDZEM or ASCZEM, again according to the recording code.
Set ADDPAR to MNSADD.

L2: decrement END; set PTR equal to END; and test the last digit of the
field: if it is zero, store ZROCOD in FILLER; otherwise, add ADDPAR to
the ASCII digit and save the result in FILLER. Call STOFIL to place the
overpunched code back in the field.

VR - If there is variable recording specified on the MTF and on the field
trailing zeroes and leading blanks are suppressed: on upper, lower and
alpha fields, the field is scanned from the end for blanks. For each
blank END is decremented. On numeric and signed numeric fields, the
field is scanned from the left.

For each zero, BEG is incremented. In both cases, when a non-zero or
non-blank character is found, control goes to TRANSF. When BEG = END
(field completely, blank or zero), control goes to MRKCHK.

Fill character (performed only if variable recording was not specified)
is done by first checking for a fill character in the format. If there is
not one, control goes to TRANSF. If there is one, the character is saved
in FILLER. IFON is called; then leading zeroes in N and S fields and
trailing blanks in A, L and U fields are replaced with the fill character
through successive calls to STOFIL.

TRANSF - Starting with BEG, load a character to A with GETCH, save it in
ATT/CURCH, store it in the TRB using SBYTE, increment ATT/RCNT and BEG
and repeat. When BEG = END + 1, continue to MRKCHK.

MRKCHK - If variable recording is specified, transfer a field separator
code to the TRB (if the recording code is BDC, use a special field
separator); and increment ATT/RCNT.

Return to caller.

Subroutine IFON.

Flags a format in the table at the end of the EOD1 buffer as one requiring
a call to RSQ.

Subroutine GETCH - loads the character indicated in Q from the input buffer
(SST/STEMP + 2 indicates the address of the first data word in the record)
Subroutine STOFIL - stores the character in FILLER back into the input
buffer at the position indicated by PTR.

INTERFACE

1. ENTRY METHOD

RSQ is an overlay of the WBT supervisor function, called by ENTPRG when
requested by MTFMON.

2. INPUT PARAMETERS OR CONDITIONS

I = current ATT address
ATT/INBUF contains the address of the WBT input buffer
ATT/REL contains relative address within the input buffer of the
record to be processed.
ATT/MTCA + 1 contains program index of caller.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ RSQ - Resequencing Program _____ Level 6___

3. EXIT METHOD

   Release the program and load next overlay by calling ENTPRG with
   Q = (ATT/MTCA + 1).
   A = 0 for a normal exit
   A = 3 for an abort exit

4. OUTPUT PARAMETERS OR CONDITIONS

   I = current ATT address
   The processed record is contained in the TRB

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      SST ($47)

   b. SST

      CFFA (12)
      STEMP (20)
      SBYTE ($95)
      GFCORE ($9D)
      ENTPRG ($9E)
      EXDFUN ($9F)
      RELFOR ($AD)
      CAM ($AE)
      WAIT ($B0)

   c. ATT

      CURCH (2)
      RCNT (4)
      INBUF (5)   - base address of input buffer
      ATRB (6)
      PFBA (7)
      REL (8)   - relative address of start record within input buffer
      FMTNO (13)
      RFA (14)
      MTCA (46)
      ATEMP (73)
      NFLDS (ATEMP +1)
      CHRPTR (ATEMP + 2)

    d. Mag tape communication area (MTCA)

        EODADR (MTCA + 2)
        MGT (MTCA + 3)

    e. CONSTANTS

ASCSEP	($1C)	Field separators
BCDSEP	($23)	
ASCZEP	($7B)	overpunch constants
ASCZEM	($7D)	
BCDZEP	($3F)	
BCDZEM	($21)	
PLSADD	($10)	
MNSADD	($19)	
EODLNG	(40)	EOD buffer length

2. PROGRAM DEPENDENCIES

SBYTE (GSE) - Stores a character into the TRB.
GFCORE - Loads a format into memory.
ENTPRG - Called to pass control back to MTFMON
EXDFUN (ENTPRG) - Releases the program.
RELFOR (GFCORE) - Releases the TRB area.
CAM - Called to request a larger TRB
WAIT (ENEXSR) - Delays processing for one clock cycle.
SETMOV (ENEXSR) - Allows program to move.
PRTMSG (SUPER) - Prints memory full error message.

3. DATA AREAS USED

ATT - active terminal table
TRB - terminal record buffer
INBUF - WBT input buffer
MTCA - mag tape communication area
EOD1 - format stack
MTF - mag tape format description

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$187_{16} = 391_{10}$.

2. RELOCATION TYPE

Run anywhere.

3. ENTRANCY CLASSIFICATION

Not reentrant.

ANALYTICAL SUPPLEMENT

The EOD1 system label buffer has ten extra words at the end for a format
stack. The stack works as follows:

For each record processed, MTFMON scans the last nine words in the stack for
a matching format. If the format number is not in the stack yet it is placed
in the next available entry (the first of the ten words points to the latest
entry in the stack). When RSQ is called, and one of the RSQ functions is
required for this format, bit 15 on the stack entry is set. Otherwise, the
entry is left as it is. Next time MTFMON searches the stack for this format,
if bit 15 is set, RSQ is called; otherwise, the call to RSQ is skipped. If
the stack overflows (more than nine formats in a WBT call), the stack is
effectively cleared and started over.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ RTB - Return to Basic Mode and Return to Document   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

The RTB function is called whenever an exit was made from Entry mode
to Read mode or from Verify mode to Read mode, and the operator wants
to return to the previous basic mode. ATT is restored and pointers
are updated to reflect the old status (possibly changed during READ).
It also returns control to the last document used.

2. DETAILED DESCRIPTION

RTB makes itself movable in memory, by the SETMOV routine. Word 2
of ATT is inspected for pure READ mode (bit 9 = 1, all the other bits
8 to 15 are 0). If pure read, next step is at OK. Otherwise, check
if current mode is Entry (bits 8-9 = 3). If not, an error message is
issued (return to document is allowed only in Entry mode). When return
to document was intended, SW1-B10 is checked. If zero (current record
is already part of the document), an error message is issued. Same
error occurs if character count in current record (ATT - W4) is not 0
(not beginning of record), or if the document number (ATT - W17) is 0.

RTD: When return-to-document function is detected, the current format
(ATT - W23) is obtained through the GFCORE routine. Bits 10 and 3 of
SW1 are zeroed. If current format is inactive, bit 3 is set again, the
format released (by the RELFOR routine), the format number, document
number and beginning-of-field are displayed on the header and the RTB
function is terminated by passing control to CLRSR. Else, the new
document and format numbers are displayed, and exit is to CLRSR.

OK: If here, the RTB instruction was OK. SW1 is checked and if necessary
(bit 13 is set) the record is written. The disk address of the basic
mode record/is transferred to the RTB and the record is read. The field
count is cleared, the relative address to the first field is set to 4
(ATT - W14). If a search buffer was assigned, the search buffer is
released (by RELFOR), If the actual format is not the basic mode format,
the last one is read into core, the current format field address (in SST)
is set to point to the last field definition in FMT (the point where an
exit to READ was made). The character count in current record (ATT - W4)
is set to the character count in basic mode (ATT - W15) and if it is 0,
control passes to BOR (beginning of record). Otherwise, a loop (UP) checks
the current position in the record against the format fields (could be
changed while in READ mode). The ATT - W14 is incremented accordingly
each time a field is checked; for the last field in format (ending after
the current position) the character count is updated to the difference
between current position and beginning of that field.

AA5641   PRINTED IN THE USA

If it is 0, BOR receives control. Else, SW2 - bit 13 is reordered.
If no counters are needed, go to BOR. If they are needed, the record
count (ATT - W4) is set to include current field (if in middle of field,
record count points now after it). The character count in current field
(ATT - W3) is set to the maximum field length. The 2 counters' signs
(obtained from FMTPAR) are reordered and the current field value is
added (or subtracted) from the corresponding counters, so the inverse
action at revalidation time (at the field's end) will keep the counters
at their correct value. The field counter and record counter are restored.

BOR: By the STTYPE routine, the data type for current field is set in
ATT. If basic mode is verify or record count is 0 then SW2 - B9 is
cleared, and control passes to CONT. Else, if in Entry mode and the
record count is not 0, the next free disk address is set to point to
the current disk address so the complete record will be written in the
same place as the old one. SW2 - B4 is set and bit 9 is checked. When
set, control passes to CONT. Else, the available number of words in
track (ATT - W8) is increased by the current record length + 5.

CONT: SW2 - B11 is checked. If not, SETIDK receives control. Else, if
the record count is not 0 or basic mode is Verify, go to OUT. Record
count 0 and basic mode Entry, causes Sw1 - B2 to be set.

SETIDK: The IDK bit (ATT-W2-B14) is set.

OUT: The current mode (ATT-W2-B14) is set to the basic mode. If it was
Read of Entry, exit through DISP (and by REGEN the screen is regenerated
according to the ATT) and then CLRSR. If current mode is Verify, then
ATT - W24 is set to the value of ATT - W2 (indication for VMM2 if
regenerating is required or not) and if no forward motion is needed
(end of batch) exit as in ENTRY. Otherwise, EXDFUN releases the RTB
function and control passes to VMM2 to continue verify.

INTERFACE

1. ENTRY METHOD

    RTB is entered by MPC through ENTPRG.

    Calling RTB module from disk.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

3. EXIT METHOD

If not returning to entry, exit to VMM2 after EXDFUN, and search next field to verify.

If Entry, reset screen to old status and exit to dispatcher.

If not returning from READ or illegal document number, exit to ERROR.

4. OUTPUT PARAMETERS OR CONDITIONS

(I) unchanged.

ATT updated to reflect the current mode (basic mode, possibly effected by changing formats while in READ).

SW1 and SW2 changed accordingly.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ONEBIT ($23)
        NZERO ($12)
        ZROBIT ($33)
        ZERO ($22)
        SST ($47)

    b. ATT equates

        ATEMP (73)
        CMODE (2)
        FCNT (3)
        BMDA (17)
        DOCNO (17)
        RFA (14)
        BRCNT (15)
        ASRCH (28)
        CFFA (12)
        SW1 (19)
        RCNT (4)
        CURDA (9)
        DOCFMT (7)
        BFMTNO (7)
        PFBA (7)
        SWITCH (24)
        STEMP (20)
        SW2 (10)
        ATRB (6)
        NDA (11)
        AVLTK (8)
        FMTNO (13)
        BMODE (15)

     c.   TRANSFER VECTOR

         EXDFUN ($9F)
         ERROR ($96)
         REWRIT ($A4)
         READN ($A6)
         CLRSR ($93)
         GFCORE ($9D)
         RELFOR ($AD)
         DISP ($A0)

2.   MACROS USED

     None.

3.   PROGRAM DEPENDENCIES

     a.   EXTERNALS

         SETMOV
         VMM2
         DBOF
         STTYPE
         DDOC
         COUNT
         REGEN
         DFMT
         FMTPAR

     b.   TRANSFER VECTOR

4.   DATA AREAS USED

     ATT
     FBA
     SST
     FMT
     TRB
     Search Buffer

PHYSICAL CHARACTERISTICS

1.   MEMORY SPACE REQUIREMENTS

     $D4_{16} = 212_{10}$ words.

2.   RELOCATION TYPE

     Run anywhere, loader, relocatable.

3.   ENTRANCY CLASSIFICATION

     Reentrant.

DOCUMENT CLASS ___SCANV___

PROGRAM NAME___ Scan Interrogate Buffer for next character or number ___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    SCANV has 3 entry points:

    GETALP extracts the next character from the Interrogate buffer.

    GETBIN extracts characters up to a comma or EOR, and converts them to a binary number.

    GETCOM scans the Interrogate buffer for a comma, EOR, or non-space character.

2. DETAILED DESCRIPTION

    a. GETALP

        1. (GETALP). Get next input character in A and Q registers.

            ●   If character is EOR ($FF) exit to caller with (A) = $00FF and (Q) = $FFFF.

        2. (CHECK). Mask off bit 7 of character.

            ●   If character code is not in the range $20 - $5F, increment SST/STEMP by 1 (character count) and exit to caller with (A) = $FFFF and (Q) = $FFFF.

            ●   If character is a comma ($2C), increment SST/STEMP by 1 (character count), and exit to caller with (A) = 0 and (Q) = 0.

            ●   For any other code in the range $20 = $5F, increment SST/STEMP by 1 and exit to caller with (A) = input character code and (Q) = 1.

    b. GETBIN

        1. (GETBIN). Set SST/STEMP + 2 = 0 (Binary result)

        2. (LOOP). Call GETALP to get next input character.

            ●   If next character is a comma, exit to caller with (A) = Binary result, and (Q) = 0.

            ●   If next character is EOR, exit to caller with (A) = Binary result, and (Q) = $FFFF.

            ●   If next character is not a valid ASCII character, exit to caller with (A) = $FFFF and (Q) = $FFFF.

222

DOCUMENT CLASS ___SCANV___

PROGRAM NAME___ Scan Interrogate Buffer for next character or number ___

3. (CONVT). If next character is a space ($20), call GETCOM to scan interrogate buffer for EOR. If a non-space character intervenes, exit to caller with (A) = $FFFF and (Q) = $FFFF. If not, exit to caller with (A) = Binary result and (Q) = $FFFF.

4. (NOTSPC). If next character is not in the range 0-9 ($30-$39), exit to caller with (A) = $FFFF and (Q) = $FFFF. Add next digit to binary result as follows: 10X (Binary result) + (next digit) → Binary result. If overflow occurs (Binary result exceeds 32, 767) exit to caller with (A) = $FFFF and (Q) = $FFFF. If not, repeat from step 2.

c. GETCOM

1. (CMLOOP). Call GETALP to extract the next character from the Interrogate buffer.

- If it is a comma, exit to caller with (A) = 0 and (Q) = 0.

- If it is not a valid ASCII character, exit to the caller with (A) = $FFFF and (Q) = 1.

- If it is EOR, exit to caller with (A) = $00FF and (Q) = $FFFF.

- If it is a space ($20), repeat step 1.

- Otherwise, exit to caller with (A) = next character and (Q) = 1.

INTERFACE

1. ENTRY METHOD

a. GETALP

```
EQU    GETALP ($93)
RTJ    GETALP
---Return---
```

b. GETBIN

```
EQU    GETBIN ($99)
RTJ    GETBIN
---Return---
```

c. GETCOM

```
EQU    GETCOM ($9C)
RTJ    GETCOM
---Return---
```

2. INPUT PARAMETERS

(I) = Base Address of Active Terminal Table (ATT)
SST + STEMP = Character Count in Interrogate Buffer.

223

DOCUMENT CLASS ___SCANV_____

PROGRAM NAME____Scan Interrogate Buffer for next character or number_____

3. EXIT METHOD

    a. GETALP

       (A) = $FFFF and (Q) = $FFFF   Error: Non-ASCII character

       (A) ='$FF and (Q) = $FFFF   EOR

       (A) = 0 and (Q) = 0   Comma

  $20 $\leqslant$ (A) <$60 and (Q) = 1   (A) = ASCII Character

    b. GETBIN

       (A) = $FFFF and (Q) = $FFFF   Error: Non-numeric character, or binary value not in the range 0 - 32767.

       (A) $\geqslant$ 0 and (Q) = $FFFF   EOR: (A) = Binary number

       (A) $\geqslant$ 0 and (Q) = 0   Comma: (A) = Binary number

    c. GETCOM

       (A) = $FFFF and (Q) = 1   Error: Non-ASCII character

       (A) = $FF and (Q) = $FFFF   EOR

       (A) = 0 and (Q) = 0   Comma

  $20 < (A) < $60 and (Q) = 1   Non space character

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

       LPMASK (2)
       TEN ($46)
       SST ($47)

    b. SST

       STEMP (20)

    c. ATT

       AINT (27)

2. MACROS USED

  None

3. PROGRAM DEPENDENCIES

  None

224

DOCUMENT CLASS ___SCANV___

PROGRAM NAME___Scan Interrogate Buffer for next character or number___

4. DATA AREAS USED

    SST   System Status Table
    ATT   Active Terminal Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$52_{16} = 82_{10}$ words

2. RELOCATION TYPE

Core resident, Loader Relocatable

3. ENTRANCY CLASSIFICATION

Reentrant

225

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____SDMY - Supervisor Dummy Function_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    SDMY is the dummy program for supervisor functions.

2. DETAILED DESCRIPTION

    SDMY jumps to SUPER (REJSUP) with A register containing error code
    33 (undefined command).

INTERFACE

1. ENTRY METHOD

    SDMY is a disk resident program loaded by ENTPRG after a supervisor function
    has been requested.

    ENTPRG finds out that the program to execute this function is not on disk and
    a call is made to load SDMY.

2. INPUT PARAMETERS OR CONDITIONS

    None.

3. EXIT METHOD

    Program exits by calling REJSUP (SUPER) with no return.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    SST
    REJSUP ($B2)

2. PROGRAM DEPENDENCIES

    SUPER (REJSUP calls PTTMSG to print error message).

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE

    7_{10} words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___SET - Set a Counter_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

SET processes the SET, M, XXX...XS command, where M is a batch counter number from 1 to 7, XXX...X is a value up to 15 digits long, and S is a + or - sign. (If the sign is missing it is assumed to be a positive number).

The value is packed, 4 digits to a word, in the corresponding batch counter area in the Active Terminal Table (ATT).

2. DETAILED DESCRIPTION

a. (SET). Call GETALP to get the 4th character in the input string.

- If the character is not a comma, go to step f.

- Call GETBIT to extract the next input field and convert it to a binary number.

- If the field is not terminated with a comma, go to step f.

- If the number is not in the range 1-7, go to step f.

- Use the number to calculate the relative position of the batch counter in ATT.

- If the current value of the designated batch counter is not zero, go to step f.

b. (NEXTC). Call GETALP to get next input character.

- If comma or an invalid character was input, go to step f.

- If End of Record was detected, go to step g.

- If the character is not a sign, go to step d.

- If plus sign was input, go to step c.

- If minus sign was input, set value of bits 12-15 of first word in batch counter to $F.

c. (CEOR). Call GETALP to get field terminator character.

- If field is terminated by End of Record, go to step g.

- If not, go to step e.

d. (CHECON). If the character is not a digit, go to step e.

- Insert the digit into lowest 4-bit field of batch counter, shifting other digits one position to the left.

- If there was not an overflow to the M.S.P., repeat from step b.

e. (EXITER). Set the batch counter to zero.

f. (ERR). Set ATT/SW1/8 = 1 (Reset must regenerate display), and exit
    to ERROR with (a) = $13 (ILLEGAL PARAMETER).

g. (EXIT). If value of batch counter is zero, go to step f.

    ● Call DISP to regenerate display.

    ● Exit to CLRSR.

INTERFACE

1. ENTRY MODE

    SET is a disk-resident program which is called via ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

    (I) = Base address of Active Terminal Table (ATT).

3. EXIT METHOD

    Normal exit is to CLRSR.
    Error exit is to ERROR.

4. OUTPUT PARAMETERS OR CONDITIONS

    (I) = Base address of ATT.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        CLRSR ($93)
        ERROR ($96)
        GETBIN ($99)
        GETALP ($9A)
        DISP ($A0)

    b. SST

        STEMP (20)

    c. ATT

        SW1 (19)
        COUNT (36)

PRINTED IN THE USA

    d. EXTERNALS

       REGEN   Parameter for DISPLY to regenerate the display.

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

ENEXSR	(CLRSR)	Clear Exits stack and go to dispatcher.
GSE	(ERROR)	Display error code and terminate the function.
DISPLY		Regenerate the display.

4. DATA AREAS USED

| SST | System Status Table |
| ATT | Active Terminal Table |

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $58_{16} = 88_{10}$ words.

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant code.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ SKIPKY - Skip Key _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   SKIPKY is a disk resident routine, loaded and executed when skip key
   is entered.

2. DETAILED DESCRIPTION

   SKIPKY calls first SETMOV to clear the program header I/O bit.
   SKIPIT is called to process automatic filling for auto fields and
   checks if SKIP key is legal when accepted. On return, further actions
   are taken according to the mode and the position in the record.

   a. Read mode:

       If not at end of record BOFFE is called for updating to next field.
       Field count is set to zero and next field is displayed. If end of
       data has not yet been reached, SKIPKY exits through CLRSR; else it
       exits to ERROR with error code $7.

       If End of Record, check if record should be written to disk and if so,
       call REWRIT (in PIO). If basic mode is Entry and End of Data was
       reached exit to SETNFM. If not yet EOD, read next record via READN
       and if different format, get new format, set pointers to format
       field address, set data type in ATT, and display new field.

       If basic mode is Verify check if last in batch and exit to ERROR
       if so, else get next record and continue as in case of basic mode
       = Entry.

   b. If not Read mode, and right zero fill is specified in format, special
       display is done according to parameters set by CALCDS in SKIPIT.
       After the function is released by EXDFUN, SW1 bit 9 is checked. If
       it is set the function will exit to EMMSKP, else SKIPDY exits according
       to mode: In Entry and Pseudo Entry to EMM1 (in EMM).
       In Verify and Pseudo Verify to ENDOF1 (in VUM).

   Internal subroutine:

   TSTEOD - Tests end of data in Read mode, by comparing number of records to
            number of records in read, and no. of characters in record to no.
            of char. read in record.

            On exit A=0 indicates end of data.

AA5641

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___SKIPKY - Skip Key_____ Level 6

INTERFACE

1. ENTRY METHOD

    SKIPKY is loaded via ENTPRG. It has only one entry point SKIPKY.

2. INPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

3. EXIT METHOD

    If Read mode - if end of data =

                    RTJ-   (EXDFUN)
                    JMP+   (SETNFU)
                    if not end of data -
                    JMP-   (CLRSR)

    If not Read and SW1-B9 is set -   RTJ-   (EXDFUN)
                                       JMP+   EMMSKP

    If Pseudo Entry or Entry-

                                       RTJ-   (EXDFUN)
                                       JMP+   EMM1 (entry point in EMM)

    If Pseudo Verify or Verify-

                                       RTJ-   (EXDFUN)
                                       JMP+   END OF 1 (entry point in VMM)

4. OUTPUT PARAMETERS OR CONDITIONS

    I = current ATT base address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        GFCORE ($9D)
        CLRSR ($93)
        REWRIT ($A4)
        DISPLY ($A0)
        READN ($A6)
        ERROR ($96)
        EXDFUN ($9F)
        SST ($47)

    b. SST

       CFFA (12)

    c. ATT

       SW1 (19)
       SW2 (20)
       BRCNT (15)
       RCD (32)
       RCDRD (31)
       FMTNO (13)
       ATEMP (73)
       ATRB (6)
       PFBA (7)
       RFA (14)

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

EMM	-	EMM1 -	EMM entry point for validation at End of Field.
		EMMSKP -	Set IDK and SKIP conditions.
		BOFFE -	Updates FFA and RFA, increases field no., and puts data type in the ATT.
		STTYPE -	Extracts data type from format and puts it into ATT.
SKPDUP	-	SKIPIT -	Processes automatic filling for alpha and numeric fields, and updates pointers for data.
SETMOV	-		Clears the program I/O bit, and sets relative bit for addressing.
DISPLY	-	DSKUP -	Display manual SKIP/DUP Entry/Verify Mode
		DSKPR -	Display SKIP Read Mode.
GFCORE	-		Load a new format to core.
CLRSR	-		Clear stack and exit to dispatcher.
PIO	-	REWRIT -	Rewrite record to disk.
		READN -	Read next record in batch.
		SETNFM -	Set SW2 - G11 for 'no forward motion'.
ERROR	-		Display error message at terminal.
EXDFUN	-		Release disk resident function.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __SKIPKY - Skip Key__ Level 6

4. DATA AREAS AND SYSTEM TABLES USED

SST
ATT
FMT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$71_{16} = 113_{10}$ words.

2. RELOCATION TYPE

Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

AA5641

PRINTED IN THE USA

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    SKPDUP has 7 entry points:

    SKIPIT blank fills or zero fills a variable field according to data
    type and user option. If a fixed field has not been filled, a diagnostic
    message is displayed. SKIPIT also supplies a plus sign (+), if no sign
    is entered in a signed numberic field.

    DUPIT adjusts counters in the Active Terminal (ATT) to point to the end
    of the current field.

    RELIT advances through the remaining fields in the current format,
    performing end of field validation as required for each field.

    VALID is the entry point used by IDKCH if the RELEASE key is used at
    the end of a boundary check field.

    TSTEOR is used by ERRKEY when automatic error flagging is performed
    during the process of completing a record with the RELEASE key.

    EOBIT executes the End of Batch own code routine specified by the
    user; then returns to the EOB procedure.

    CALODS calculates the current address in the Terminal Record Buffer (TRB),
    and the number of words remaining in the current field.

2. DETAILED DESCRIPTION

    a. SKIPIT

        1) Call ENTSR to save return address in exit stack.
            If ATT/SW1/2 = 1 (first character of record is next)
            call BORENT to perform beginning of record processing.

        2) If ATT/MODE/12 = 1 (Pseudo Verify mode), go to step 19.

            If ATT/MODE/13 = 1 (Pseudo Entry mode), go to step 3.

            If ATT/MODE/9 = 0 (Verify mode), go to step 19.

            If ATT/MODE/8 = 0 (Read mode), go to step 15.

        3) (ENTRY). If keying mode is variable, go to step 6.

            If ATT/FDTP/13 = 0 (not signed numeric field), go to step 4.
            If positioned at last character in current fields, go to step 5.

        4) (ERRORF). Set Q = 4 and call ERROR to display error code A4
            (Illegal Function key) and exit.

        5) If current character is not a sign, go to step 6. If not
            boundary check field, go to step 6. Toggle ATT/SW1/9 (skip
            key must be next).

6) (VARY). If ATT/FDTP/13 = 0 (unsigned numeric field) go to step 8. If ATT/CURCH/0-7. If not, set ATT/CURCH/0-7 = plus sign (+).

7) (STORE). Call SBYTE to store sign character in Terminal Record Buffer (TRB). Increment ATT/FCNT by 1 (character count in current field), increment ATT/RCNT by 1 (character count in current record). If positioned as first character of field, go to step 8. Call DISPLY to display current column and character.

8) (NOSIGN). If field is full, call EXTSR to exit. If ATT/FDTP/15 = 0 (field type =Lower or Alpha), go to step 9. If ATT/FDTP/14 = 0 (field type = Upper), to to step 9. If blanks in numeric field and at beginning of field, go to step 9. Otherwise (Field type = Numeric or Signed numeric) use zero fill character ($30) and go to step 10.

9) (ALPHA). Use blank fill character ($20).

10) (TEST). Save fill character in ATT/ATEMP = 1.

AA5641   PRINTED IN THE USA

Set ATT/ATEMP + 2 = 0 (Fill character count).

Set ATT/MISCH = 0 (Mismatched character), clear IDK bit, and if fixed keying field, check for signed numeric. If ATT/SW2/15 = 0 (no nonzero characters entered yet), go to step 22. If blank fill, go to step 20. If auto function = R or L, go to step 22. If ATT/FDTP/13 = 0 (Field type = Lower, Alpha, Upper, or Numeric) go to step 21. If ATT/FDTP/13 = 1 (Field type = Signed Numeric) go to step 22.

20) If Auto Function = N, D, S, or F, go to step 22.

21) (ERF). Go to step 4.

22) (TSTLOP). Calculate number of characters left in field. If field full, go to step 23. If not full, go to step 24. If character count exceeds field capacity, set ATT/SW1/9 = 0 (skip need not be next), and call EXTSR to exit.

23) If ATT/FDTP/13 = 0 (Field type = lower, Alpha, Upper, or Numeric), go to step 24. Set ATT/CURCH/0-7 = plus sign. Save plus sign in ATT/ATEMP + 1.

24) (TSL1). Call GBYTE to get next character from input string. If it matches go to step 26. If ATT/SW1/5 set, move back to beginning of field and go to step 21. If blanks in numeric field specified, go to step 21.

25) (OUT). Set (Q) = ERRAS5 and call ERROR to display error code A5 (mismatched character) and exit.

26) (INCRF). Increment ATT/FCNT, and ATT/ATEMP + 2 by 1 and repeat from step 22.

27) (LEFTZ). If ATT/FCNT, = 0 (Field empty), go to step 12. Save ($7FFF- (ATT/FCNT/0-7) + 1) in SST/STEMP + 1 (number of characters in field in overflow counter form). Save (ATT/RCNT) - (ATT/FCNT/0-7 + 1) in ATT/ATEMP + 3 (First character position in field in SST/STEMP + 2 and SST/STEMP + 3. Decrement ATT/RCNT by 1 (Character count in record). Save ATT/MODE in ATT/ATEMP + 2.

28) (MOVLOP). Align field to right. Call GBYTE to get last character in input buffer. Call SBYTE to store the character in right-most position of field. Adjust input and output pointers one position to the left and repeat from MOVLOP until all characters entered in the field have been moved to the right.

29) (CONT). Set ATT/MODE = fill character.

30) (UP). Insert fill character to the left of entered character until the field is full. Restore ATT/MODE and set ATT/RCNT to end of field. Call EXTSTR to exit.

b.

1) Set SST/STEMP = Current address in Terminal Record Buffer (TRB).

   Set SST/STEMP + 1 = $8000 if current byte position is bits 8-15.

   Set SST/STEMP + 1 = $0000 if current byte position is bits 0-7.

2) Set ATT/ATEMP + 3 = Current address in TRB. Exit to caller.

   Set ATT/ATEMP + 4 =

```
      15   14              0
      ┌───┬────────────────┐
      │ P │       N        │
      └───┴────────────────┘
```

where:

    P (bit 15) = 0 if current byte position is in bits 0-7.
             = 1 if current byte position is in bits 8-15.

    N (bits 0-14) = Number of character positions remaining in field.

c. DUPIT

1) Call Caldos to calculate current address in Terminal Record Buffer (TRB) and number of character positions remaining in current field.

2) Advance ATT/RCNT and ATT/FCNT to end of current field. Exit to caller.

d. RELIT

1) Call SKIPIT to finish processing current field.

e. VALID

1) Set ATT/SW2/6 (release in operation).

2) If ATT/MODE/9 = 0 (VERIFY Mode) go to step f1.

3) Call VALDTE for end of field processing (nonzero test, limit check, special test, own code, batch counter update).

4) Continue at step f1.

f. TSTEOR

1) Clear ATT/SW1/15 (Cont needed reset) and ATT/SW2/15 (List nonzero flag).

2) If last field in format, go to step 8. If not, call BOFFE to update SST/FFA, ATT/RFA, ATT/FLDN, and ATT/DT for next field.

3) If ATT/BMOD/13 = 1 (Entry mode) go to step 6. If not, call CHKVRY to check if current field is verifiable. If not verifiable, go to step 5. If sight verify, set ATT/SW1/5 = 1. If not verifiable, go to step 5.

4) Set ATT/FCNT/0-7 = 0 (Character count in field). Continue at step d1.

5) (NOTV). Update ATT/RCNT to end of current field. Repeat from step 1.

6) (ENTRY 1). Call AUTO to perform auto skip/dup, if called for. Continue at step 1.

7) (ENDREC). Clear ATT/SW2/6, if ATT/MODE/9 = 0 (Verify mode) exit to EORTST. If ATT/MODE/9 = 1 (Read or Entry mode) exit to TSTODD.

g. EOBIT

1) Call ENTPRG to load and execute End of Batch own code.

2) Call ENTPRG to load and execute EOB.

INTERFACE

1. ENTRY METHOD

SKIPIT, DUPIT, AND CALSDS are called by the RTJ instruction. RELIT, VALID, TESTEOR, and EOBIT are entered by JMP instructions.

2. INPUT PARAMETERS OR CONDITIONS

(I) = Base address of Active Terminal Table (ATT).

3. EXIT METHOD

SKIPIT returns to the location immediately following the RTJ SKIPIT except when an error is detected, in which case SKIPIT exits via ERROR. DUPIT and CALCDS exit to the location immediately following the RTJ which called them. RELIT continues processing at VALID. VALID continues processing at EORTST. EORTST exits to EORTST in Verify mode, to TSTODD in Read or Entry mode. EOBIT exits to EOB procedure.

4. OUTPUT PARAMETERS OR CONDITIONS

    (I) = Base address or ATT.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        ENTSR ($91)
        EXTSR ($92)
        CLRSR ($93)
        GBYTE ($94)
        SBYTE ($95)
        ERROR ($96)
        ENTPRG ($9E)
        DISPLY ($A0)

    b. SST

        CFFA (12)
        FFA (12)
        STEMP (20)

    c. ATT

        CMODE (2)
        MODE (2)
        CURCH (2)
        FCNT (3)
        RCNT (4)
        MISCH (5)
        ATRB (6)
        CURDA (9)
        FDTP (13)
        BMOD (15)
        BRCNT (15)
        SW1 (20)
        RCRD (31)
        RCD (32)
        ATEMP (73)

d. ASCII CODES

    PLUS ($2B)
    MINUS ($2D)

e. ERROR CODES

    ERRA4 ($4)
    ERRA5 ($5)
    ERRB2 ($7)

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

    ENTSR   Save return address in exit stack
    EXTSR   EXIT TO LAST ADDRESS IN EXIT STACK
    CLRSR   Clear exits stack and function locks
    GBYTE   Get current character from Terminal Record Buffer (TRB).
    SBYTE   Store character in TRB
    ERROR   Display Error code and exit
    ENTPRG   Enter a disk resident program
    DISPLY   Display a message
    BORENT   GET A TRACK FOR NEXT RECORD
    VALDTE   End of field validation
    BOFFE   Advance pointers to next field
    CHKVRY   Check for current field verification
    AUTO   Auto skip/duplication

4. DATA AREAS USED

    SST   System status table
    ATT   Active terminal table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $164_{16} = 452_{10}$ words.

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____SOF - Supervisor Off Command_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    SOF is a supervisor function to exit from the supervisor function
    dispatching program SUPER before the time delay is over.

2. DETAILED DESCRIPTION

    a. If not end of record after SOF command exit to REJSUP.

    b. Moves basic input logical unit ATT/39 to current input logical
        unit ATT/36. Moves basic list logical unit ATT/40 to current
        list logical unit ATT/37.

    c. Calls SUPRW to output message "SV OFF" to supervisor console.

    d. If not terminal 0 (TTY), clear supervisor bit ATT/0/7. Call
        DISPLY at entry point SETDIS to set display definition bits in
        TOPT to IDLE (10) and exit to CLRSR. If TTY simply exit to CLRSR.

INTERFACE

1. ENTRY METHOD

    SOF is a disk resident program which is called via ENTPRG.

2. INPUT PARAMETERS

    I = current ATT address.

3. EXIT METHOD

    Jump to REJSUP with A = 9 if not end of record after SOF.

    Jump to CLRSR (ENEXSR) is normal exit.

4. OUTPUT PARAMETERS OR CONDITIONS

    For consoles other than 0 (TTY) the supervisor bit ATT/0/7 is
    cleared and the display definition bits in the TOPT are set to
    10 (IDLE).

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___SOF - Supervisor Off Command___ Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK ($2)
        ZROBIT ($33)
        SST ($47)

    b. SST

        STEMP (20)
        CLRSR ($93)
        GETCOM ($96)
        SUPRW ($97)
        DISPLY ($A0)
        REJSUP ($B2)

    c. CONSTANTS

        FW (3)

2. PROGRAM DEPENDENCIES

    PIO SUPRW to output message to supervisor.

    SCANV GETCOM Get comma, EOR or nonspace char. from input string

    SUPER REJSUP supervisor reject procedure.

    ENEXSR CLRSR common completion return.

    DISPLY SETDIS to set display definition bits in TOPT.

3. DATA AREAS USED

    SST system status table

    ATT active terminal table(for basic and current logical units)

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $20_{16} = 32_{10}$ words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____SPCKEY - Space Key_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    SPCKEY, called whenever the space bar is pressed in Verify or Entry
    modes returns to MPC with a zero or a space, depending on the data
    type. In Read mode, ATT pointers and the display are advanced to the
    next character in a record. This may result in an advance to the next
    field or the next record.

2. DETAILED DESCRIPTION

    SETMOV is called, allowing the program to be moved in the FRP. The
    function is checked for legality: if pure Read mode, jump to RED. If
    IDK -start of record (entry)
    (ATT/MODE/14 = 1 and ATT/SW1/2 = 1), call BORENT to initialize the
    record.

    If IDK - sight verify, go to FLDOK in the Read Mode Code.
    (ATT/MODE/14 = 1 and ATT/SW1/5 = 1)

    If IDK-Mismatch (ATT/MISCH ≠ 0), clear ATT/MISCH, and clear the IDK
    bit.

    If any other IDK condition is present, exit to ERROR with Q = 4
    (A4 - illegal function).

    When not Read mode, the data type is checked: if the data type is L,
    A or U, a blank ($20) is stored in ATT/CURCH/0-7. Otherwise a zero
    ($30) is saved. EXDFUN is called to release the program and control
    jumps to the CHR entry point in MPC.

    Read Mode (label RED): if ATT/FCNT is zero, subroutine TSTEOD is
    called to test for end-of-data. If A = 0 on return, control goes to
    ERROR with Q = 7 (B2-end of data).

    FLDOK: ATT/FCNT and the format are checked to see if the current position
    is at the end of a field. If not, ATT/RCNT and ATT/FCNT are incremented
    and control passes to DSPCAL.

    Otherwise if this is sight-verify, the program exits to ERROR with Q = 4.
    (Can't space past end-of-field in sight verify).

    For a normal Read mode end-of-field condition, ATT/RCNT is incremented.
    The format is checked to see if this was the last field in the record.
    If not, BOFFE is called to advance pointers to the next field, ATT/FCNT
    is set to 0, and TSTEOD is called.

If A is nonzero on return, control goes to DSPCAL. Otherwise, the program exits to ERROR with Q = 7.

For an end-of-record condition, RCNT is set to zero. If a rewrite is required REWRIT is called.

ATT/RCRD is incremented. If the basic record count (ATT/BRCNT/0-11) is nonzero, control goes to GETNXT. Otherwise, if the Basic mode is Entry TSTEOD is called. If A = 0 on return, EXDFUN is called and control passes to SETNFM (in PIO). If the basic mode is Verify, if this is the last record in the batch (TRB/5 is negative) exit to SETNFM.

GETNXT: Read the next record and check if the format of the new record is the same as that of the previous record. If not, call GFCORE to read down the new format. Set up the RFA and CFFA parameters to point to the first field of the format. Call STTYPE to store the correct data type in the ATT. FCNT is cleared, TSTEOD is called again and control goes to DSPCAL.

DSPCAL: DISP is called with ADC DSPACE (display space key in Read Mode). The program exits to CLRSR.

Subroutine TSTEOD: end-of-data is tested:

a. Basic mode Verify: if CURDA, CURDA + 1 equal the address of the next record (TRB/6,4) then this is end-of-data.

b. Basic mode Entry: If ATT/RCD = ATT/RCDRD and ATT/BRCNT = ATT/RCNT then this is end-of-data.

    On exit, A = 0 indicates end-of-data.

INTERFACE

1. ENTRY METHOD

    SPCKEY is called by a jump from ENTRPG after ENTPRG was called by MPC with function key code $20.

2. INPUT PARAMETERS

    (I) = base address of active terminal table (ATT).

3. EXIT METHOD

    Entry, Verify: exit to CHR in MPC.

    Read, Sight Verify: exit to CLRSR (ENEXSR).

    On error conditions (illegal function or end-of-data): exit to ERROR (GSE).

AA5641

4. OUTPUT PARAMETERS OR CONDITIONS

    On exit to ERROR, Q = 4 or Q = 7.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        CLRSR ($93)
        ERROR ($96)
        GFCORE ($9D)
        EXDFUN ($9F)
        DISP ($A0)
        REWRIT ($A4)
        READN ($A6)

    b. SST

        CFFA (12)

    c. ATT

        MODE (2)
        CURCH (2)
        FCNT (3)
        RCNT (4)
        MISCH (5)
        ATRB (6)
        PFBA (7)
        CURDA (9)
        DT (13)
        FMTNO (13)
        RFA (14)
        BMOD (15)
        BRCNT (15)
        SW1
        SW2 (20)
        RCDRD (31)
        RCD (32)

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME _____ SPCKEY - Space Key _____ Level 6

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   ERROR   -   Exit taken when error detected
   GFCORE -   Replaces current format with that of a new record
   CLRSR   -   Clears enter/exit stack on program exit
   EXDFUN -   Drops user count on a function
   READN   -   Reads next record in a batch
   REWRIT -   Rewrites current record to disk
   DISP   -   Central entry for all display calls
   DSPACE -   Index to the display routine for space key in Read mode
   BORENT -   Called to initialize a new record
   CHR   -   Entry in MPC taken for normal Entry/Verify
   SETMOV -   Subroutine that clears I/O bit on the program and sets bit on
               the function index
   SETNFM -   Entry in PIO taken at end-of-data condition in Read or Verify
   BOFFE   -   Subroutine in EMM that advances pointers to the next field
   STTYPE -   Sets field type into the ATT for the current format

4. DATA AREAS USED

   ATT, TRB, FMT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $90_{16} = 144_{10}$ words.

2. RELOCATION TYPE

   Run anywhere, disk resident.

3. ENTRANCY CLASSIFICATION

   Reentrant.

AA5641

DOCUMENT CLASS SPILL

PROGRAM NAME Active Batch Table handling routine

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    SPILL performs 4 operations on the Active Batch Table (ABT):

    Locate a designated ABT entry and bring it into core,

    Update a designated ABT entry,

    Delete a designated ABT entry,

    Insert a new ABT entry.

2. DETAILED DESCRIPTION

    a. (SPILL). If (FLOCK) ≠ 0 (SPILL function in use), exit to STRTRY to schedule caller's 'Busy' return address.

        ● If not, set (FLOCK) = $FFFF, set (NXTB) = $FFFF (next batch number), set (DISTB) = $4000

        ● Call ENTSR to save return address in Exits stack.

        ● Get core address of Active Batch Table (ABT) and disk address of SPILL table from Extended System Table (EST).

        ● If requested function is 'Get ABT Entry' go to step b

        ● If requested function is 'Update ABT Entry' go to step f

        ● If requested function is 'Insert ABT Entry' go to step h.

    b. (GETABT). Call GETENT to locate ABT entry for designated job. Set (FLOCK) = 0. If entry not located, set (OPCODE) = 0 and continue at step d

    c. (SPILL 2). Set OPCODE/0-7 = entry number in ABT, and OPCODE/8-15 = Relative address on disk of ABT entry for designated job.

    d. (SPILL3). Call RTNSR to extract the return address from the Exits stack.

        ● If (FLOCK) ≠ 0, go to step e.

        ● Move ABT entry to caller's parameter area.

    e. (NOTRNS). Set (FLOCK) = 0 and exit to caller with (A) = (OPCODE) and (Q) = (NXTB).

    f. (UPDABT). Calculate address of designated entry in ABT: Address = 3 x Entry Index + ABT Address.

        ● If designated entry is not currently in core, call ABTIO to read it from disk.

        ● Clear bits 0-2 of Batch track address (word 2 of ABT entry).

DOCUMENT CLASS ___SPILL___

PROGRAM NAME___Active Batch Table handling routine___

- If mode (Bits 13-15) of update value is maintenance, save old mode in bits 0-2 of Batch track address.

- Store update value in word 0 of ABT entry.

- If update value ≠ 0, go to step i.

- If update value = 0 (Delete entry), decrement count of entries by 1.

g. (UPD22). If ABT sector is not empty, go to step i.

- If empty ABT sector is not last sector in ABT, or is the only sector, go to step i.

- If empty ABT sector is the last sector, but not the only sector, call ABTIO to read the preceeding sector from disk. Set last sector flag (Bit 15 of word 2 of sector), and repeat tests from step g.

h. (INSABT). Save new entry parameters and call GETENT to scan ABT for duplicate entry.

- If a duplicate entry was located, go to step d with (A) = $FFFF.

- If ABT sector currently in core is not the first ABT sector, call ABTIO to read first ABT sector from disk.

- Call GETENT to locate first empty ABT entry.

- If no empty entries, go to step d with (A) = 0.

- Move new entry parameters to empty ABT entry.

- Increment count of entries by 1.

i. (INSAB3). Call ABTIO to write current ABT sector to disk.

- Go to step b

INTERFACE

1. CALLING SEQUENCE

```
          EQU     SPILL ($A1)
START     ENQ     F                    Function Code
          RTJ-    SPILL
          ADC     X
          ADC     Y
          ADC     Z
          JMP*    START
          --(Return)--
          F = 0, Get an ABT entry.
```

          X = Job Number
          Y = Batch number (if Y = $FFF, Function is to check if job is active)
          Z = undefined

DOCUMENT CLASS SPILL

PROGRAM NAME Active Batch Table handling routine

F = 1, Update ABT entry.

    X = ABT entry address (in same format as ATT/ABTN)
    Y = Update value to replace word 0 of entry
       (If Y = 0, Function is to delete the entry)
    Z = undefined

F = 2, Insert ABT entry

    X,Y,Z = 3 word ABT entry to be inserted.

(I) = Base address of Active Terminal Table (ATT).

2. OUTPUT PARAMETERS OR CONDITIONS

    (A) = 0, Error, requested entry not found (F = 0),
       or no space available for insertion (F = 2).

      < 0, Error, duplicate entry in ABT (F = 2).

      > 0, (A) = address of requested entry (F = 0, X,Y, and Z contain entry
             parameters), or address of inserted entry (F = 2).

    (Q) > 0, next batch number in ABT (A = 0 and F = 0).

    (Q) = \$FFFF, no greater batch number exists in ABT (A = 0 and F = 0).

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        KXFF (\$A)
        KXFFF (\$E)
        NZERO (\$12)
        ZERO (\$22)
        ONEBIT (\$23)
        MINUSI (\$33)
        SST (\$47)
        ENTSR (\$91)
        RTNSR (\$98)
        REDSEC (\$A7)
        STRTRY (\$AF)

    b. SST

        ATT (11)
        RPINCR (SST + 13)
        ABTADR (22)
        SPIL (23)

220

DOCUMENT CLASS __SPILL__

PROGRAM NAME __Active Batch Table handling routine__

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

	ENTSR	Save return address in Exits stack
ENEXSR	RTNSR	Retrieve return address from Exits stack
	STRTRY	Schedule retry address
PIO	(REDSEC)	Read a sector from disk

4. DATA AREAS USED

SST   System Status Table

5. EXTERNALS

SISPLE
SIABTW
SIABTS

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$14C_{16} = 332_{10}$ words

2. RELOCATION TYPE

Core resident, loader relocatable

3. ENTRY CLASSIFICATION

Reentrant. Partially locked.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME_____ SRCH1 _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The program is an interrogate function and the first overlay of the search function. It performs syntax analysis of the serach request, saves the parameters in the search buffer, and transfers control to the next overlay.

2. DETAILED DESCRIPTION

    The operator is allowed to issue the request only in Read mode and the legal command formats are:

$$\text{SRN,} \begin{Bmatrix} S \\ F \\ B \end{Bmatrix} \text{,rr} \begin{bmatrix} Fn \\ \text{,}Cn \end{bmatrix}$$

$$\text{SER,} \begin{Bmatrix} S \\ F \\ B \end{Bmatrix} \begin{bmatrix} Fn \\ \text{,}Cn \end{bmatrix}$$

$$\text{SDM,} \begin{Bmatrix} S \\ F \\ B \end{Bmatrix} \text{, [ff]} \text{ , } \begin{Bmatrix} Fn \\ Cn \end{Bmatrix} \text{,M , } \begin{bmatrix} \begin{Bmatrix} Fn \\ \text{,}Cn \end{Bmatrix} \text{,M} \end{bmatrix}$$

where:
- S — start
- F — forwards
- B — backwards
- rr — number of records to be skipped
- Fn — field number n. $0 < n < 128$
- Cn — column number n. $0 < n < 1024$
- ff — format number
- M — character string with 10 character at most
- { } — one possibility should be chosen
- [ ] — optional

The "enable termination" bit is set and the search buffer is released if existent. The mode is checked to be pure Read mode. If not, the program exits through the common error exit with "illegal request" error code.

A search buffer is built for SRN or SER as in Appendix A.

The search buffer is built for SDM as in Appendix B.

The following checks are made on the parameters:

a. Search direction is S, F or B.

b. If the request code is SRN a record number different from 0 should be specified.

c. If an identification field number is specified, it should be less than 128.

    If an identification column number is specified it should be less than 1024.

AA5641

PRINTED IN THE USA

d. If the request code is SDM, and a format number is specified, it should be
    an existing format in the system. The sup bit is set and the format is
    brought into core by GFCORE. If a mask is specified with column counter
    the column counter should be less than the record length specified in the
    format.

    If a field number corresponds to a mask, a format should be specified and
    the field should be specified in the format.

    If 2 sequential commas (,) appear in a data mask, a single comma is entered
    to the character string.

    If a field identifier is specified in the command whose data type is numeric,
    leading zeroes are ignored in the mask string.

    If any of these checks fails, the progrogram exist via the common error exit
    with appropriate error code; invalid format no. if the error is connected
    with the format, or illegal parameter otherwise before exit if a new format
    has been brought to core (detected by the supervisor bit), the old format
    whose number is specified in the TRB is brought to core by GFCORE.

    Common error exit:

    The supervisor bit which was set if format has been brought to core is
    cleared. The "enable termination" bit is cleared. The search buffer
    is released if existent. The regenerate bit in SW2 is set and the
    program exits to ERRØR with the appropriate error code.

    If no syntax error is detected the program exits through the normal exit.

    The interrogate buffer is released and the supervisor bit is cleared.
    The second overlay is then entered.

INTERFACE

1. ENTRY METHOD

    The program is loaded via INTRGT after the operator issued a search
    command: SRN, SER or SDM.

2. INPUT PARAMETERS OR CONDITIONS

    A search command (as appears in the program description) string is
    in the interrogate buffer. The system locore temporary SST+TEMP1 ($5B)
    points to the character following the 3 request code characters in the
    interrogate buffer.

3. EXIT METHOD

    a. Normal Exit

    The program releases itself and enters the next overlay via ENTPRG

```
              RTJ-    (EXDFUN)
              LDQ     =N$2808
              CLR     A
              RTJ-    (ENTPRG)
```

    b. Error Exit

    The program exits to ERROR with appropriate error codes.

4. OUTPUT PARAMETERS OR CONDITIONS

    a. Normal Exit

    The search buffer holds the parameters which were specified in the search command as described in Appendices A, B.

    The terminate bit is set.

    The interrogate buffer backward pointer in the ATT is 0.

    b. Error Exit

    Possible error codes:

    1) $12   -   Illegal request

    2) $13   -   Illegal parameter

    3) $E   -   Invalid format number

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

    LPMASK (2)
    NZERO ($12)
    ZERO ($22)
    ONEBIT ($23)
    ZROBIT ($33)

    b. SST

    SST ($47)
    TEMP1 (20)
    TEMP2 (21)
    TEMP3 (22)
    TEMP4 (23)
    IRBPOS (20)   -   position in interrogate buffer for SCANV routines

    c. TRANSFER VECTOR

        ERROR ($36)
        EXDFUN ($SF)
        ENTPRG ($9E)
        GFCORE ($9D)
        CAM ($AE)
        GETALP ($9B)
        GETBIN ($99)
        WAIT ($B0)

    d. ATT

        IRB (27)
        SRB (28)
        TRB (6)
        FBA (7)
        MODE (2)
        SW1 (19)
        CURDA (9)
        CURN (31)
        RKDRD (31)
        TRMNT (23)
        FMTNO (13)
        ATTMPP (73)

    e. ERROR CODES

        ERR1 ($12)
        ERR2 ($13)
        ERR3 ($E)

    f. ASCII CODES

        B ($42)
        C ($43)
        F ($46)
        M ($4D)
        N ($4E)
        R ($52)
        S ($53)

2. MACROS USED

   None.

AA5641

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____SRCH1_____ Level 6

3. PROGRAM DEPENDENCIES

    a. Entry points:

       SRCH1

    b. External list and routines called via transfer vector:

          ENEKSR $\begin{cases} \text{SETMOV} \\ \text{WAIT} \end{cases}$

          GSE $\{$ ERROR

          ENTPRG $\begin{cases} \text{EXDFUN} \\ \text{ENTPRG} \end{cases}$

          GFCORE $\{$ GFCORE

          CAM   - Get user search buffer to hold the parameters

          SCANV $\begin{cases} \text{GETALP} \\ \text{GETBIN} \end{cases}$

          FMTSCN - find parameter list for the identification field

4. DATA AREAS AND SYSTEM TABLES USED

    SST
    ATT
    IRB - interrogate buffer
    TRB
    SRB - user search buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    Program length - 184_{16} words.
    User SRB   - 8 or ^{16}20 words.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS **CYBERDATA IMS**

PROGRAM NAME **SRCH1**   Level 6

ANALYTICAL SUPPLEMENT

Working storage layouts: See Appendix A and B for the layout of user
search buffer (SRB).

AA5641

APPENDIX A | 1 |

```
     15         12          8          4          0
    +-----------------------------------------------+
    |            BACKWARDS POINTER                  |  0
    +-----------------------------------------------+
    |                 LENGTH                         |  1
    +-----------------------------------------------+
    |               USER COUNT                       |  2
    +-----------------------------------------------+
    |               TRB LENGTH                       |  3
    +-----------------------------------------------+
    |                    0                           |  4
    +----+-----------------------------+------+------+
    | F  |  FIELD/COLUMN NUMBER-1       | C    | F    |  5
    | C  |                             | M D  | S B  |
    +----+-----------------------------+------+------+
    |             RECORD NUMBER                      |  6
    +----+-----------------------------+------+------+
    | F  |  FIELD COLUMN NUMBER        | C    | F    |  7
    | C  |                             | M D  | S B  |
    +----+-----------------------------+------+------+
    |                                                |
    +-----------------------------------------------+
    |                                                |
    +-----------------------------------------------+
    |                                                |
    +-----------------------------------------------+
    |                                                |
    +-----------------------------------------------+
    |                                                |
    +-----------------------------------------------+
    |                                                |
    +-----------------------------------------------+
    |                                                |
    +-----------------------------------------------+
    |                                                |
    +-----------------------------------------------+
     15         12          8          4          0
```

AA5641   PRINTED IN THE USA

DOCUMENT CLASS   CYBERDATA IMS

PROGRAM NAME   SRCH1   Level 6

2

```
15         12          8          4          0
┌──┬─────────────────────────────┬────────┬────────┐
│F │                             │  C     │  F     │
│  │   FIELD/COLUMN NUMBER -1     │   M    │   S    │
│C │                             │    D   │    B   │
└──┴─────────────────────────────┴────────┴────────┘
```
5

BIT 0-1   -   search
        00   -   start
        01   -   forwards
        10   -   backwards

BIT 2-3   -   request type
        00   -   SRN
        01   -   SER
        10   -   SDM

BIT 4-14   -   field or column column number - 1

BIT 15   -   field or column
        1   -   field
        0   -   column

```
15         12          8          4          0
┌────────────────────────────────────────────────┐
│                                                  │
│              RECORD NUMBER                        │
│                                                  │
└────────────────────────────────────────────────┘
```
6

If the request is SRN,S the record number is saved as - (record number - 1)
record number is number of records to be skipped

AA5641   PRINTED IN THE USA

3

```
15         12          8          4          0
┌──────────────────────────────────────────────┐
│                                                │
│                                                │       7
│                                                │       ──
└──────────────────────────────────────────────┘
```

Word 7 is 0 if no identification field or column is specified for SER, or equal to word 5 in the other cases.

In the SRCH2 overlay when the request code is SRN word 7 is the record skip count.

```
15         12          8          4          0
┌──────────────────────────────────────────────┐
│                                                │
│                                                │       ──
│                                                │
└──────────────────────────────────────────────┘
```

AA5641   PRINTED IN THE USA
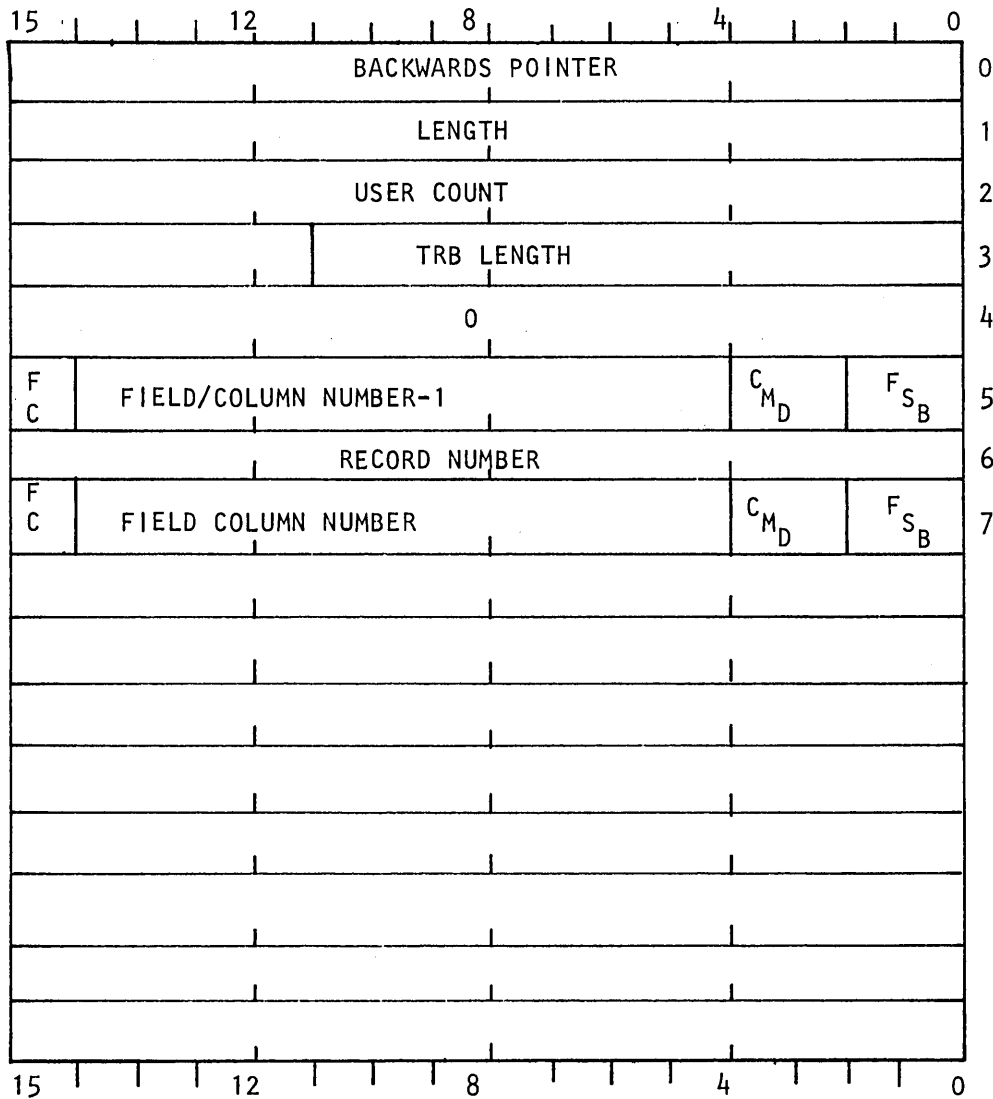
DOCUMENT CLASS _____ CYBERDATA IMS

PROGRAM NAME _____ SRCH1   Level 6

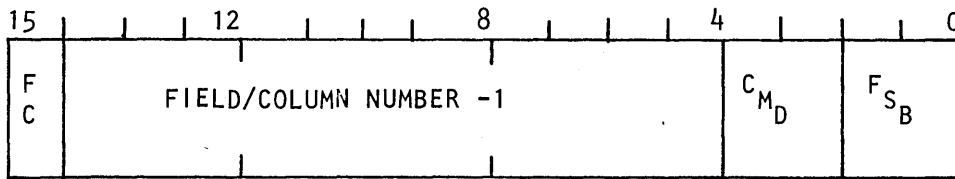APPENDIX B   SRCH1

```
      15 |   |   | 12 |   |   |   8 |   |   |   4 |   |   |   0
     ┌──────────────────────────────────────────────────────────┐
     │              BACKWARDS POINTER                             │  0    ____
     ├──────────────────────────────────────────────────────────┤
     │                    LENGTH                                  │  1
     ├──────────────────────────────────────────────────────────┤
     │                  USER COUNT                                │  2
     ├──────────────────────────────────────────────────────────┤
     │                  TRB LENGTH                                │  3    ____
     ├──────────────────────────────────────────────────────────┤
     │                 FORMAT NUMBER                              │  4
     ├──────────────────────────────────────────────────────────┤
     │                                    C M       F S          │  5
     │                                      D         B          │
     ├──────────────────────────────────────────────────────────┤
     │   FIELD LENGTH     │  LENGTH OF FIRST DATA MASK           │  6
     ├──────────────────────────────────────────────────────────┤
     │              COLUMN NUMBER            │  C     │ D        │  7
     │                                       │        │ T        │
     ├──────────────────────────────────────────────────────────┤
     │                   DATA MASK                                │  8
     ├──────────────────────────────────────────────────────────┤
     │                                                           │  9
     ├──────────────────────────────────────────────────────────┤
     │                                                           │  10
     ├──────────────────────────────────────────────────────────┤
     │                                                           │  11   ____
     ├──────────────────────────────────────────────────────────┤
     │                                                           │  12
     ├──────────────────────────────────────────────────────────┤
     │   FIELD LENGTH     │  LENGTH OF SECOND DATA MASK          │  13
     ├──────────────────────────────────────────────────────────┤
     │              COLUMN NUMBER            │  C     │ D        │  14
     │                                       │        │ T        │
     ├──────────────────────────────────────────────────────────┤
     │                   DATA MASK                                │  15   ____
     └──────────────────────────────────────────────────────────┘
      15 |   |   | 12 |   |   |   8 |   |   |   4 |   |   |   0
```

AA5641

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___ CYBERDATA IMS

PROGRAM NAME ___ SRCH1   Level 6

APPENDIX B   SRCH1
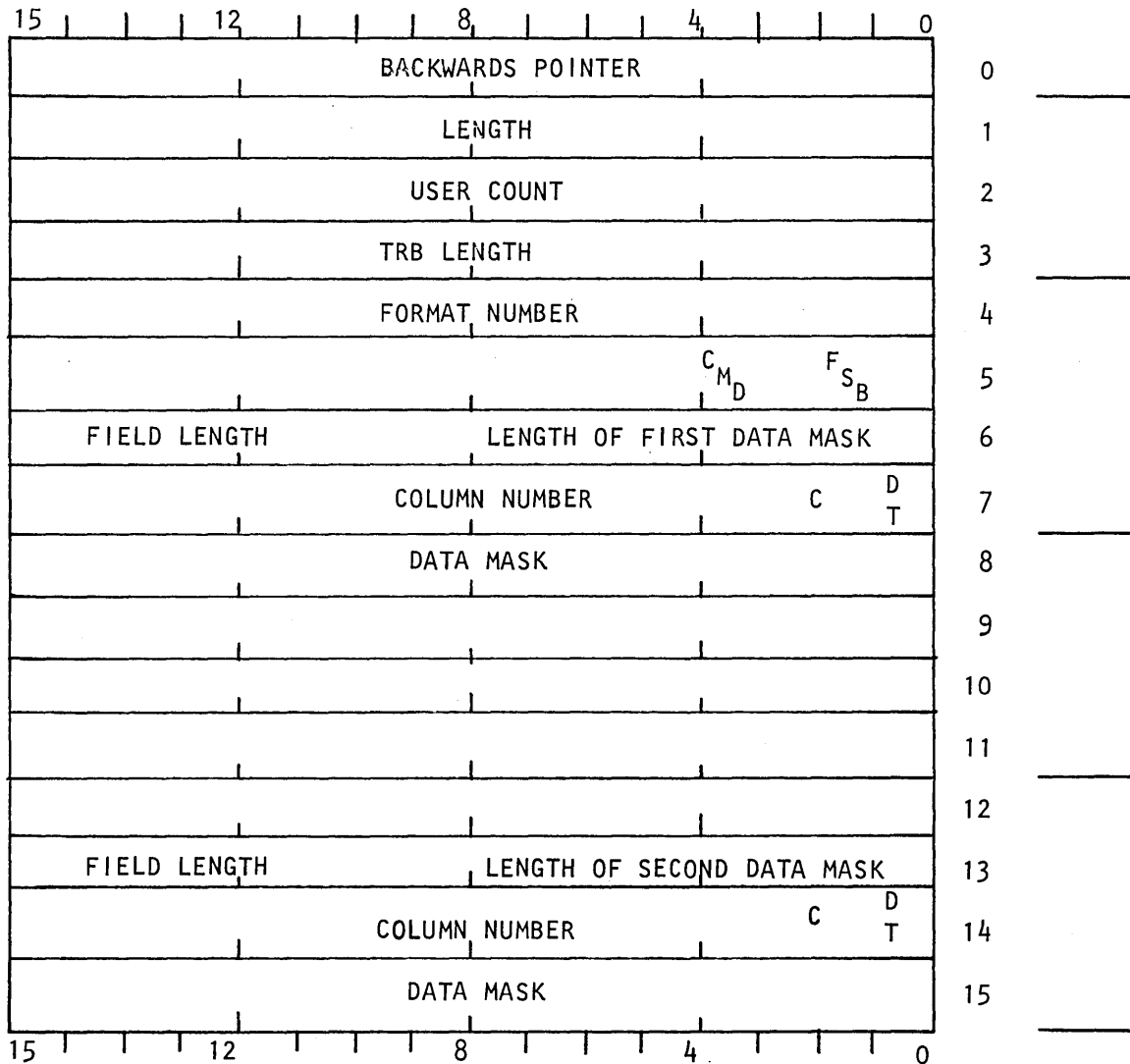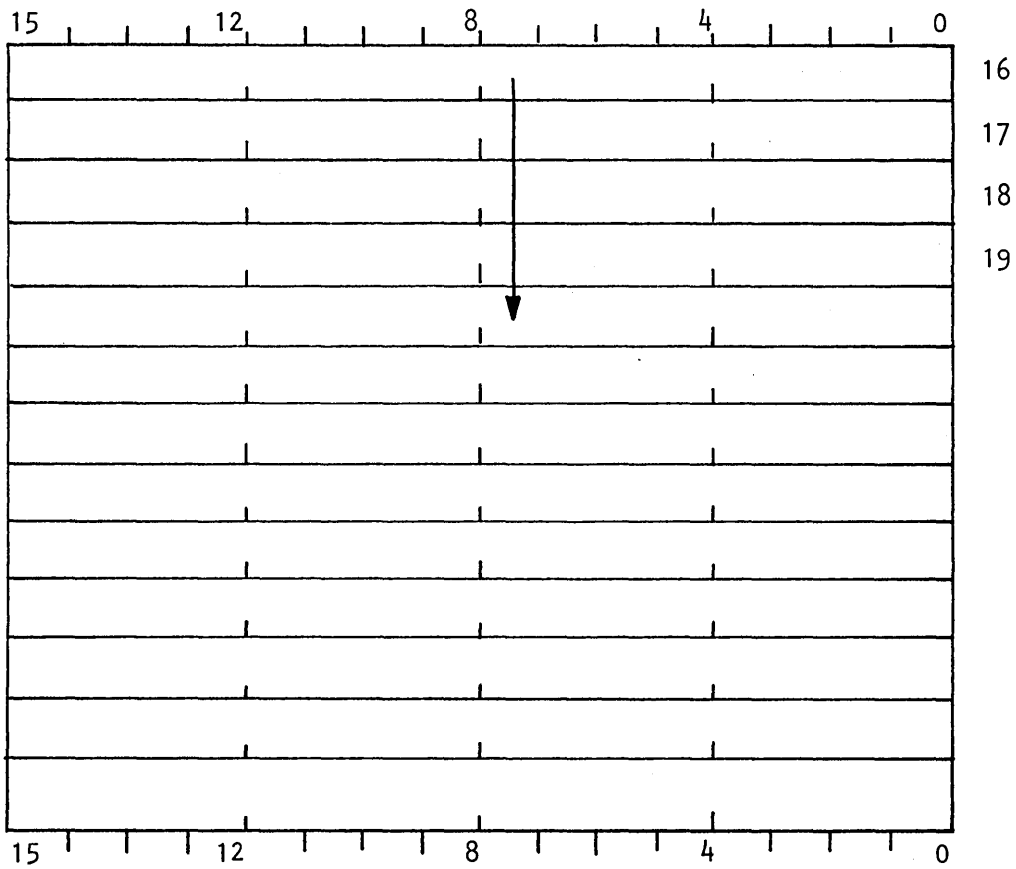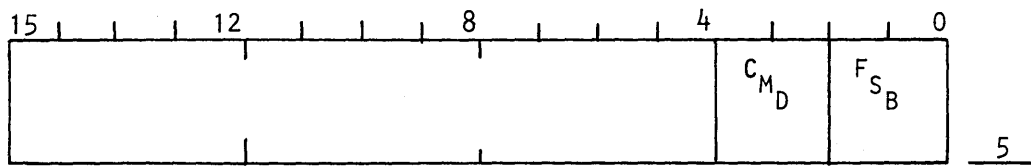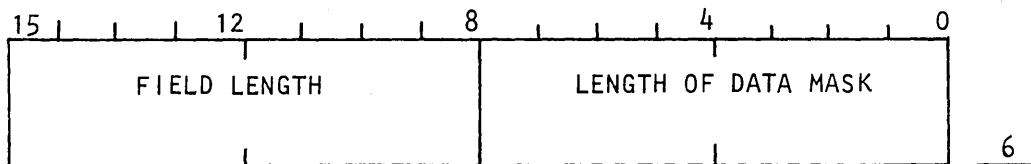
AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___SRCH1___ Level 6

APPENDIX B SRCH1

```
15    |   |   |  12  |   |   |   8  |   |   |   4  |   |   |   0
┌──────────────────────────────────────────┬───────┬───────┐
│                                            │  C    │  F    │
│                                            │   M   │   S   │
│                                            │    D  │    B  │      5
└──────────────────────────────────────────┴───────┴───────┘
```

    BIT 0-1   -   search direction
         00 -   start
         01 -   forwards
         10 -   backwards

    BIT2-3   ---   request type
         10 -   SPM

```
15  |   |   |  12  |   |   |  8   |   |   |  4   |   |   |  0
┌───────────────────────────┬───────────────────────────────┐
│      FIELD LENGTH          │       LENGTH OF DATA MASK      │
│                            │                                │      6
└───────────────────────────┴───────────────────────────────┘
```

    BITS 6 -7   -   length of first data mask, up to 10 characters.
    BITS 8-15   -   length of the masked field of FC bit (bit 15)
                     in word 7 is 0.

               -    0 otherwise

AA5641
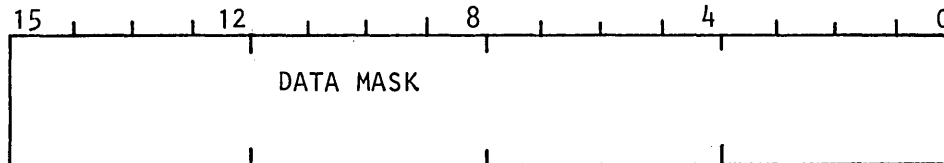
DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___SRCH1_____ Level 6

APPENDIX B

SRCH1

```
 15         12          8          4          0
┌──────────────────────────────┬────┬───┬───┐
│                              │    │   │ D │
│        COLUMN NUMBER         │    │ C │ T │      7
│                              │    │   │   │
└──────────────────────────────┴────┴───┴───┘
```

BIT 0 - data type

   1 -   non numeric
   0 -   numeric

BIT 1 - field or column mask

   1 -   column
   0 -   field

       data type is specified only for field mask

BIT 4-15   from which character in record the masking should be performed

```
 15         12          8          4          0
┌──────────────────────────────────────────────┐
│                                                │
│                  DATA MASK                     │      8 - 12
│                                                │
└──────────────────────────────────────────────┘
```

UP to 10 ASCII characters

AA5641

PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME ___SRCH2 - Search 2___   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

The program is the second overlay of the search functions. It searches
the batch to find the record specified by the parameters transferred in
the user search buffer. When the record is found it is read into the
TRB and the screen is regenerated.

2. DETAILED DESCRIPTION

If the interrogate buffer is not released the program was entered when
the SØN command was issued. The interrogate buffer is released. If no
search buffer is existent the program exits with "illegal request" error
code. Otherwise the terminate bit is set.

If the program is already used by the maximum number of users (an EQU in
the program) the program exist through the common exit with completion
code. Otherwise the "override monitor ATT check" bit is set and the TRB
is released.

The following section of the program can be used only by one user.

In this part a data buffer is allocated if not existent already and is
connected to the EST. Its length is dependent on the available free
memory. If the least requirement is not satisfied the program exits.
This buffer is used by all the users. Every user in turn reads a part
of his batch to this buffer. If the record he searches is not present
in this buffer and an additional I/O operation should be performed, the
user unlocks this section and delays the I/O for at least a cycle by
entering WAIT state.

The buffer and the program now become available for another user.

The direction of search and the part of batch to be read into the
data buffer are decided as follows:

If the request code is SRN and search direction is S the direction is
changed to F if the record is after the current record or to B if the
record is closer to the current record than to the start of batch. The
number of records to be skipped is updated and saved in the search buffer
(word 7).

AA5641

When the batch is searched from the start it is read from the start
(2nd sector of FTØB) in the first I/O operation and then it is searched
forwards. If search direction is F, the batch is read from the current
record (the record whose disk address is in CURDA in the ATT) up to the
end of current track or to the end of the buffer. (The least number of
words). If search direction is B, the batch is read from the beginning
of the current record inclusive. If the data buffer is too short, the
data is not read from the beginning of track. If search is from the
start, the first record in batch is brought into core.

The current record is checked according to the specifications in the
search buffer. If the request code is SRN, the number of records to be
skipped (forwards or backwards) are checked to be zero. If not zero,
the skip count (in the search buffer) is decremented by 1.

If the request code is SER, the error bit in the records header is
checked. If the bit is set care is taken that the whole record will
be in core. The data is scanned to find the first erroneous field.

If the request code is SDM, the record which matches the data masks is
searched. The number of records to be checked is limited to 30 in a
user turn.

The mask strings are compared character by character to the data strings,
which begin in the column specified in the command. If a field identifier
was specified in the command and its data type is numeric, leading zeroes
are ignored. The retrieval of characters is done through the subroutine
GETALP in SCANV. The user search buffer and the data buffer are connected,
each in its turn, to the interrogate backwards pointer to accomplish the
requirements of SCANV.

If the current record (its disk address is in CURDA in the ATT) does not
fill the requirements, the next or previous record, depending on the
search direction becomes the candidate for checks. If the End of Batch
or Beginning of Batch are reached, the program exits via the common exit
with appropriate completion code. Otherwise the CURDA is updated and the
checks continue. If the current record is not present in core the user
enters WAIT state. The I/O operation is delayed to the next search
user's cycle.

If the current record satisfies the search requirements the program exits
via the common exit with a completion code.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___SRCH2 - Search 2___

COMMON EXIT: EØS

This exit can be reached due to several conditions.

a. The number of program users exceeds the number allowed. C.C.=1

b. There is no core for data buffer. C.C.=2

c. End of Data or Beginning of Data has been reached before a record which fills all the requirements, has been found. C.C.=3,4

d. The function has been terminated. C.C.=5

e. The record which meets all the requirements, has been found. C.C.=6

The following operations are done at the common exit:

If the program exits, not because the program is too occupied, a check is done if this is the only user. If so the search data buffer is released.

A memory buffer is allocated for the TRB and is connected to the ATT. The current record is read into the TRB according to the disk address in CURDA. The format specified in the last record is brought to core. The "monitor override check" bit is cleared. The FMTNØ in ATT is updated. If the search completion codes lack memory or there are too many users the program exits through the error exit. The FCNT, RCNT, RFA and DT (data type) in ATT and CFFA in SST are updated to match the first field in record.

If the completion code is End of Data or Beginning of Data the program exits through the error exit.

If the identification field or column was specified it is checked through FMTSCN that the field or column are valid for the format of the record. If not, the error exit is taken. Otherwise the FCNT, RCNT, RFA, DT in ATT and CFFA in SST are updated to match the identification field or column. If the search code is SDM the first identification field is used.

If the search direction is S it is now changed to F direction for SØN which may be issued. The display is regenerated, the internal clock is cleared, the program is released and exit to dispatcher is done via CLRSR.

ERROR EXIT: The display is regenerated, the user search buffer is released, the internal clock is cleared, and the program exits via ERROR with error code messages which match the search completion code.

AA5641

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___SRCH2 - Search 2_____ Level 6

INTERFACE

1. ENTRY METHODS

    a. The program is loaded and executed by INTRGT when the operator
        issues SØN command.

    b. The program is loaded and executed by SRCH1.

2. INPUT PARAMETERS OR CONDITIONS

    The user search buffer holds the parameters which were specified
    in the search command. (See Appendices A and B of SRCH1). If the
    program is loaded through INTRGT, the interrogate buffer is not released.

3. EXIT METHOD

    a. Normal Exit

        The program releases itself and exit to the dispatcher via CLRSR

                    RTJ-   (EXDFUN)
                    JMP-   (CLRSR)

    b. Error Exit

        The program exits to error with appropriate error codes.

4. OUTPUT PARAMETERS OR CONDITIONS

    a. Normal Exit

        The search buffer remains as on entrance.

    b. Error Exit

        Q register holds the error code.

    c. Possible Error Codes

        $12 -   E1 -   Illegal request (SØN was issued when no search
                                        buffer present)
        $18 -   F5 -   Function busy try again
        $17 -   F4 -   Core full try again
        $7   -   B2 -   End of data
        $6   -   B1 -   Beginning of data
        $13 -   E2 -   Illegal parameters
        $0   -   no error code to be displayed when function terminated

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___SRCH2 - Search 2___ Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)

    b. SST

        SST ($47)
        TEMP1 (20)
        TEMP2 (21)
        TEMP3 (22)
        TEMP4 (23)
        CFFA (12)

    c. TRANSFER VECTOR

        ERROR ($96)
        EXDFUN ($9F)
        GFCORE ($9D)
        DISPLY ($A0)
        CLCDWA ($90)
        GETALP ($9B)
        READN ($A6)
        CAM ($AE)
        WAIT ($B0)
        CLRSR ($93)
        ENTSR ($91)
        EXTSR ($92)
        RTRNSR ($98)
        REDDWA ($A9)
        RELFOR ($AD)
        REWRIT ($A4)

AA5641

d. ATT

FTOB (1)
FCNT (3)
RCNT (4)
TRB (6)
FBA (7)
CURDA (9)
NDA (11)
FMTNO (13)
RFA (14)
FLDN (14)
BMODE (15)
SW1 (19)
SW2 (20)
TRMNT (23)
IRB (27)
SRB (28)
CURN (31)
RCD (32)
BATCH (35)
ATTMP1 (73)
ATTMP2 (74)
ATTMP3 (75)
ATTMP4 (76)
ATTMP5 (77)

e. EST

SRCHB (16)

f. CONSTANTS

MAKUSR (10)

g. ERROR CODES

ERR0 ($12)
ERR1 ($18)
ERR2 ($17)
ERR3 ($7)
ERR4 ($6)
ERR5 ($E)
ERR6 ($13)

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   a. ENTRY POINT:

      SRCH2

   b. EXTERNAL LIST AND ROUTINES CALLED VIA TRANSFER VECTOR

   ENEXSR
   $\left\{\begin{array}{l}\text{ENTSR}\\\text{EXTSR}\\\text{RTRNSR}\\\text{CLRSR}\\\text{SETMOV}\\\text{WAIT}\end{array}\right.$

   PIO
   $\left\{\begin{array}{l}\text{CLCDWA}\\\text{REDDWA}\\\text{READN}\\\text{REWRIT}\end{array}\right.$

   - reads part of the batch
   - reads the satisfactory record

   GFCORE
   $\left\{\begin{array}{l}\text{GFCORE}\\\text{RELFOR}\end{array}\right.$

   ENTPRG   EXDFUN

   CAM   -   allocates the memory for data buffer

   DISP
   $\left\{\begin{array}{l}\text{REGEN}\\\text{SETDIS}\end{array}\right.$

   FMTSCN   -   return parameter list for the identification
                   field or column

   SCANV - GETALP
   GSE   - ERROR

4. DATA AREAS AND SYSTEM TABLES USED

   SST
   ATT
   DTL
   FMT
   SRB   -   user search buffer
   SRCHB -   program's data buffer
   TRB

AA5641

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___SRCH2 - Search 2_____Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   Program length - $2dB_{16}$ words.

   Data buffer - at least 520_{10} words.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Partially locked.

ANALYTICAL SUPPLEMENT

Working storage layouts. See Appendices A and B of SRCH1 for user
search buffer (SRB) structures.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME __STJ - Special Test Jump Table__   __Level 06__

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    STJ is a table of addresses of special test routines. The table currently holds 15 entries.

2. DETAILED DESCRIPTION

    STJ has no executable code. It consists entirely of a table of addresses of special test routines which reside in core.

INTERFACE

1. ENTRY METHOD

    No entry. Special test routines are called indirectly through the table, e.g., the calling sequence

        EXT   STJLNG
        LDQ   STJLNG
        RTJ-   2, Q

    calls special test number 3 (MOD = 10).

2. EXIT METHOD

    None

3. OUTPUT PARAMETERS OR CONDITIONS

    None

ASSEMBLY CHARACTERISTICS

1. EQUATES

    None

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    CDVTST (CDVNUM) Check digit validation routine.

AA5641

4. DATA AREAS USED

   None

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $10_{16} = 16_{10}$ words.

2. RELOCATION TYPE

   Core resident, loader relocatable.

3. ENTRANCY CLASSIFICATION

   N/A

DOCUMENT CLASS __ST005__

PROGRAM NAME___Numeric Field Validation___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   ST005 is special test routine number 5, which checks the first character of a field for the digit 1 or 2. Any other value results in a Special Test error.

2. DETAILED DESCRIPTION

   a. (SPT05). Calculate address of first character of current field.

   - If first character is the ASCII code for 1 or 2, exit to the caller.

   b. (ERR). If not, exit to ERROR with (Q) = CE9 (Special Test Error).

INTERFACE

1. ENTRY METHOD

   ```
   EXT     SPT05
   RTJ     SPT05
   ```

2. INPUT PARAMETERS OR CONDITIONS

   None

3. EXIT METHOD

   Normal exit to location immediately following RTJ SPT05.

   Error exit to ERROR with error code CE9 in Q-register.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      ```
      LPMASK ($2)
      NZERO ($12)
      ZERO ($22)
      ERROR ($96)
      ```

   b. ATT

      ```
      FCNT (3)
      RCNT (4)
      ATRB (6)
      ```

   c. DIGIT CODES

      ```
      X ($31)
      Y ($32)
      ```

DOCUMENT CLASS ___ST005___

PROGRAM NAME ___Numeric Field Validation___

      d. ERROR CODE

          CE9 ($9)

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

/ GSE (ERROR) Display error code

4. DATA AREAS USED

    ATT   Active Terminal Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $14_{16} = 20_{10}$ words

2. RELOCATION TYPE

    Core-resident, loader relocatable.

3. ENTRANCY CLASSIFICATION

    Reentrant

DOCUMENT CLASS ___SUP___

PROGRAM NAME___Make Key Entry Station a Supervisor Terminal___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    SUP checks whether the current terminal can assume supervisor status and, if so, sets up Supervisor status in the Active Terminal Table.

2. DETAILED DESCRIPTION

    a. (SUP). Call SETMOV to enable moving of SUP.

        ● If ATT/MODE/12-15 ≠ 0 (not Idle Mode), go to step b.

        ● If next input characters is not EOR, go to step b.

        ● If ATT/0/10 = 1 (CRT Display), go to step c.

        ● Call CLCDWA to calculate disk word address of Terminal Type Table (TTB) entry for current terminal.

        ● Call REDDWA to read TTB entry for current terminal into ATT/ATEMP + 1.

        ● If bit 7 of TTB entry = 1 (Supervisor status authorized for this terminal), go to step c.

    b. (ERR). Set ATT/SW1/8 (Reset must regenerate display), and exit to ERROR with (a) = $12 (ILLEGAL REQUEST error).

    c. (OK). Set ATT/0/7 (Supervisor bit).

        ● Clear ATT/SUPCOM - ATT/SUPCOM + 9 (Supervisor Communication Area).

        ● Exit to SV.

INTERFACE

1. ENTRY METHOD

    SUP is a disk-resident program which is called via ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

    (I) = Base address of Active Terminal Table (ATT)

3. EXIT METHOD

    Normal exit is to the Supervisor Command Processor via transfer Vector entry SV.

    Error exit is via Transfer Vector entry ERROR.

4. OUTPUT PARAMETERS OR CONDITIONS

    (I) = Base address of ATT

    (ATT/0/7) = 1 (Supervisor bit)

20

DOCUMENT CLASS ___SUP___

PROGRAM NAME___Make Key Entry Station a Supervisor Terminal___

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        CLCDWA ($90)
        ERROR ($96)
        GETALP ($9B)
        REDDWA ($A9)
        SV ($B1)

    b. SST

        STEMP (20)

    c. EST

        TTB (41)

    d. ATT

        MODE (2)
        SW1 (19)
        SUPCOM (36)
        ATEMP (73)

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    PIO   {CLCDWA   Calculate disk word address.
             {REDDWA   Read disk word address.

    GSE   (ERROR)   Display error message.

    SCANV   (GETALP) Get alpha character from input string.

    SUPER   (SV)   Supervisor Command Processor.

    ENEXSR   (SETMOV) Enable disk resident program move.

4. DATA AREAS USED

    SST   System Status Table
    EST   Extended System Table
    ATT   Active Terminal Table

21

DOCUMENT CLASS ___SUP___

PROGRAM NAME ___Make Key Entry Station a Supervisor Terminal___

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $3A_{16} = 58_{10}$ words

2. RELOCATION TYPE

    Disk resident, run anywhere

3. ENTRANCY CLASSIFICATION

    Reentrant

22

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___SUPER___

PROGRAM NAME___Dispatching program for supervisor functions___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    SUPER has four entry points: SV, REJSUP, PRTMSG and SUPTHD. SV locks out
    all but one command at a time. Additional commands are rejected with the
    message SV BUSY. When SV commands are not locked out, an interrogate buffer
    is allocated, the message SV is displayed, and a supervisor command is
    read. If an error is detected, the message SV INPUT ERROR is displayed.
    If not, the disk resident program DSKSUP is called to process the command.

    REJSUP displays the error message indicated by a code in the A-register,
    then executes SV code.

    PRTMSG reads the designated message from a message table on disk, and
    displays the message.

    SUPTHD is the thread of the supervisor read request and is checked if a
    disk error occurs.

2. DETAILED DESCRIPTION

    a. SV

        1. (SV). If ATT/0/13, 14 = 0 (Function Lock and Pseudo Function
            Lock both off), go to step 2. If not, write -SV- BUSY on supervisor
            console display and exit to Dispatcher.

        2. (SVOK). If ATT/FUNIDX/0-14 ≠ 0 (Disk resident program), call EXDFUN
            to release program area.

        3. Set ATT/TRMNT/14, 15 = 0 (current function cannot be terminated by
            Cancel key.

        4. (LOOP). Clear exits stack in ATT. Call WAIT to schedule step 5.
            (TCHK) as a retry address.

        5. (TCHK). Call GETBUF to allocate an 80-character Interrogate buffer;
            fill the buffer with $FFFF codes. Set User count = 1 in buffer
            header. If ATT/0/0-5 ≠ 0 (not terminal 0) go to step 14.

        6. (RTTY). (Terminal 0). Call SUPRW to write -SV- to Supervisor
            Console. Call SETIO to set I/O bit in Interrogate Buffer Header, so
            it won't be moved during I/O transmission. Set up FREAD request. Set
            ATT/0/14 = 1 (Function lock). Set ATT/RQ + THREAD = $FFFF. Issue
            FREAD on input unit. If input unit is not system Input Comment
            device, select delay count = $FFFE, and go to step 8.

        7. (DEL). If input unit is system Input Comment device, compute delay
            count = DELAYS x CSRATE (approximately 150 seconds).

        8. (STORIT). Save delay count in ATT/DELAYC.

        9. (DELAYL). If read request thread word = 0 (request complete) go to
            step 11. If not, decrement delay counter by 1. If delay counter = 0
            go to step 10. If not, call WAIT to delay one cycle. On return
            repeat step 9.

AA5641

DOCUMENT CLASS SUPER

PROGRAM NAME Dispatching program for supervisor functions

10. (TMOVER). Release Interrogate buffer. Set ATT/0/13, 14 = 0
(Function Lock, Pseudo Function Lock). Set ATT/RQ + THREAD = 0.
Move ATT/STDLST (Standard List LUN) to ATT/SUPLST and ATT/STDIN
(Standard Input LUN) to ATT/SUPIN. Schedule Timeout portion of
input driver and exit to Dispatcher.

11. (CAI). Set ATT/RQ + THREAD = 0. If bit 15 of request completion
code word = 0 (no error), save completion code word in ATT/SVTMP1
and go to step 16.

12. (INPER). If error detected, move ATT/STDIN (Standard Input LUN)
to ATT/SUPIN (Supervisor Input LUN) and ATT/STDLST (Standard List
LUN) to ATT/SUPLST (Supervisor List Lun). Call SUPRW to write
SV INPUT ERROR to Supervisor Console.

13. (CLRLCK). Set ATT/0/13, 14 = 0 (Function Lock, Pseudo Function
Lock), and repeat from step 1.

14. (RVLTRM). Call DISPLY to display SV at terminal. Call SUPRW to
read 24 words into Interrogate buffer. In case of read error, go
to step 12.

15. (RDOK). If completion code = 0 (Release key) go to step 16. If
completion code = 2 (Overflow) go to step 12. Otherwise call
DISPLY to clear screen, and repeat from step 1.

16. (COMEXT). Call ENTPRG to load and execute DSKSUP, with no return
to SV.

b. REJSUP

1. (REJSUP). Save message code in ATT/SVTMP1. Move ATT/STDIN (Standard
Input logical unit) to ATT/SUPIN and ATT/STDLST (Standard List
logical unit) to ATT/SUPLST. If message code <0 (no message), go
to step a.13. If ATT/FUNIDX/0-14 ≠ 0 (Disk resident program)
call EXDFUN to release the program areas. Call PRTMSG to retrieve
the designated message from disk and write it to the supervisor
console. Go to step a.13.

c. PRTMSG

1. (PRTMSG). Call ENTSR to save return address in exit stack.

2. Call CLCDWA to calculate disk word address of designated message.

3. Call GETBUF to allocate an 80-character Interrogate buffer.

4. Call SETIO to set I/O bit in buffer header to prevent movement.

5. Call REDDWA to read message into buffer.

6. Call CLIO to clear I/O bit in buffer header.

2/0

7. Call SUPRW to write message to supervisor console.

8. Call EXTSR to exit.

INTERFACE

1. ENTRY METHOD

    The entry point addresses of SV and REJSUP are in locations $B1 and $B2, respectively.

    Entry is as follows:

    SV

    ```
    EQU     SV ($B1)
    JMP-    (SV)
    ```

    REJSUP

    ```
    EQU     REJSUP ($B2)
    ENA     Message Code
    JMP-    (REJSUP)
    ```

    PRTMSG

    ```
    ENA     Message Code
    RTJ     PRTMSG
      --Return--
    ```

    In each case, (I) = Base address of Active Terminal Table

2. EXIT METHOD

    After writing an error message, REJSUP resumes processing at SV. Under normal conditions, SV loads DSKSUP to process the supervisor command. If the SV function is locked, SV exits to the Dispatcher. PRTMSG returns control to the location following the RTJ PRTMSG. In each case, (I) = Base address of ATT.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

    ```
    LPMASK (2)
    NZERO ($12)
    ZERO ($22)
    ONEBIT ($23)
    SST ($47)
    CLCDWA ($90)
    CLCDWA ($90)

    ENTSR ($91)
    ```

DOCUMENT CLASS ___SUPER___

PROGRAM NAME___Dispatching program for supervisor functions___

    EXTSR ($92)
    SUPRW ($97)
    ENTPRG ($9E)
    EXDFUN ($9F)
    DISPLY ($A0)
    REDDWA ($A9)
    RELFOR ($AD)
    WAIT ($B0)
    DISPCH ($EA)

  b. SST

    ATTB (10)
    STEMP (20)

  c. ATT

    FUNIDX (21)
    TRMNT (23)
    AINT (27)
    SUPIN (36)
    SUPLST (37)
    STDIN (39)
    STDLST (40)
    DELAYC (46)
    SVTMPI (46)
    RQ (65)
    ENTR (78)

  d. REQUEST LIST

    THREAD (2)
    RP (5)
    CP (5)

  e. GENERAL

    FR (2)
    FW (3)
    EDPGM (3)
    STACKL (8)
    LMSG (16)
    DELAYS (150)

2. MACROS USED

  FWRITE

212

DOCUMENT CLASS ___SUPER___

PROGRAM NAME___Dispatching program for supervisor functions___

3. PROGRAM DEPENDENCIES

CLCDWA	Calculate disk word address.
ENTSR	Save return address in exit stack.
EXTSR	Exit to last address in exit stack.
SUPRW	Supervisor read-write package.
ENTPRG	Enter a disk resident program.
EXDFUN	Exit disk function.
DISPLY	Display a message (Entry points - SETDIS, CLRFLD, DSUE).
REDDWA	Read disk word address.
WAIT	Schedule following address as retry address.
DISPCH	System Dispatcher
GETBUF	Allocate an Interrogate buffer.
DSKSUP	Process supervisor commands.
------	Input driver timeout error procedure.
RELFOR	Release format area.
ABSADD	To allow for addressing over 32K.

4. DATA AREAS USED

LOGIA	Logical Unit Table
PHYSTB	Physical Device Table
SST	System Status Table
ATT	Active Terminal Table

5. OTHER EXTERNALS

CSRATE	polling rate
SIREJM	address of error message table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $DB_{16} = 219_{10}$ words

2. RELOCATION TYPE

    Loader relocatable

3. ENTRANCY CLASSIFICATION

    Reentrant

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME_____THW - Thaw_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   The supervisor issues the command to reactivate a frozen batch on an
   idle station. If the batch is not frozen or resides on a deselected
   disk, or the station is not idle, the command is rejected. If the
   operator number specified in the command is different from the one
   specified in the first track of batch, a statistics record is written
   for the previous operator.

2. DETAILED DESCRIPTION

   The supervisor issues the command:

   THW,jjjjjj,bbb,ss[,000]

   where:

   jjjjjj - job name
      bbb - batch number
       ss - station number
      000 - operator number

   Last parameter is optional.

   The command is checked for valid syntax. If the job name has more
   than 6 characters, or the batch number or operator number are greater
   than 999, the program exits to REJSUP with an appropriate error code.

   The specified station is checked to be activated in system start up,
   not to be a supervisor on in interrogate mode, to be idle and not under
   function or pseudo function lock. If one of these conditions does not
   exist, the program exits to REJSUP.

   The job specified in the command is looked for in the LJD through GETLJD.
   If the station specified is a CRT, the job name is moved to the TOPT
   (output buffer).

   If the bit in the STT (bit 13 in word $48) which controls modification
   of ABT entries is set, the program waits until the bit is cleared. Then
   the bit is set and the ABT entry corresponding to this job and batch is
   looked for via SPILL.

   The logical unit of the batch (in second word of ABT entry) is checked
   to be selected. A buffer corresponding to the length of record
   specified in the LJD entry + 8 words for header is taken for the TRB.

If the batch is nonexistent or not frozen, the logical unit is
deselected, or there is not enough memory for the TRB, the bit controlling
modification of ABT entries is cleared, and the program exits with
appropriate error code to REJSUP. Otherwise the batch status in the
ABT is modified to active, and that bit is cleared.

The function index is moved and the base address of the ATT of the invoking
supervisory station is saved in a temporary (ATEMP+4) in the ATT of the
station on which the batch should be activated and the function lock bit
is set on it. Then the I-register is changed to the base address of the
activated station and processing continues.

The ATT image from the first track of batch is read by parts via SUBR
subroutine into the ATT in core after the monitor override bit was set.

If the operator number specified in the command is different from the
operator number in the statistics information in first track of batch,
a statistic record is written for the previous operator. The statistical
information is moved to words 34-69 in the ATT. The number of records
the operator entered is calculated as the difference between the number
of records in the batch and that value when the previous statistics record
of this batch was written. The key depressions, and number of verify
corrections are set to zero, and the modified ATT is written on the ATT
image. The statistic information in first track of batch is updated.
Accumulated time is put to zero, the operator number is modified, and
current number of records in the batch is updated. The start time of
run is updated to the current clock. This is done also when the operator
number has not been changed. The updated statistic information is
written to disk.

The counters area is read into the ATT.

The data status is restored. If ATT was written by FRZ routines it is
indicated by the negative record count (word 4).

If the ATT is a "frozen" ATT, the error lock is updated according to the
PFBA, where it was saved. The TRB backwards pointer is moved from the
invoking supervisory ATT. The controlling format is brought into core.
If the batch is under document control, the ELDT (word 16 in ATT, distance
to current format group) is changed from relative to absolute mode.
Display mode is changed to Data mode and the last processed record is
read into the TRB according to the current record disk address. If the
ATT is "frozen", it can correspond to End of Data state.

In this state the most significant bit of the Interrogate buffer backwards pointer should be set. If the most significant bit of ASCNT (word 5 in ATT) is set, it indicates that the ATT was written by CORKEY in an End of Data state. (Deleting the last record, or insert after the last record.) In both cases the disk address of the previous record in TRB is assigned the value of the current record disk address.

If the ATT was written by FRZ routines, the Format Change bit was saved in AINT (Interrogate buffer backwards pointer) and is reset in the TRB header. The disk address which was saved in the 8 L.S.B. of AINT and in ASRCH is moved to CURDA in ATT. This address may be different from the address of the last processed record, if the last processed record was an inserted one. Then AINT and ASRCH are cleared.

Pointers connected to the format are updated according to the RCNT. RFA, FCNT are updated. If FRZ was performed in an End of Field state, a flag ($FF) was saved in FCNT and this state is restored.

If the batch is under format control (FMTNØ≠0), the display mode is set. The screen is regenerated. The bits for "override monitor check" and 'LL' have to be displayed and are cleared.

Function index is cleared and the program is ready to exit. If the mode of the batch is Verify several cases have to be considered:

If no verification options are specified, no special key has to be entered, and it is not INSERT mode, the RCNT and FCNT are updated to point to End of Record and the address of EØRTST (End of Record Test in VMM) is saved in the enter/exit stack to be performed in the next cycle and the pseudo function lock bit is set to simulate wait state.

If verification options are specified and it is beginning of record but not End of Data, the address of VMM1 ("check verification on field" entry in VMM) is used. If the format change bit is set in TRB header the appropriate bit is set in SW1.

In all cases some bit in switch words are cleared, one register is assigned, the base address of supervisor's ATT, the MSG pointer is restored, and the function lock imposed on the supervisor is cleared. The program exits to SV.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___THW - Thaw_____ Level 6

INTERFACE

1. ENTRY METHOD

   The program is loaded and executed via DSKSUP.

2. INPUT PARAMETERS

   Supervisor command THW in the interrogate buffer.

3. EXIT METHOD

   a. Error exit is through REJSUP.

   b. Normal exit is through SV.

4. OUTPUT PARAMETERS OR CONDITIONS

   a. If exit is through REJSUP the following error codes are
      possible:

      0   Invalid job name if job name has more than 6 characters
               or does not exist in the LJD.

      4   INVALID BATCH NUMBER. Batch number not between 1 and 999,
               or batch does not exist.

      86   Invalid operator number. Operator number is not between 1 and 999.

      52   BATCH NOT FROZEN. Thaw is illegal operation.

      65-68   DISK N DESELECTED. Where N is 1-4. Batch resides on deselected
               disk.

   If the batch is in Verify, the address of EORTST or VMM1 (entry points
   in VMM) will be placed in the ENTRY/EXIT stack, and the pseudo function
   lock (ATT/0/13) will be set.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ZERO ($22)
      ONEBIT ($23)
      ZROBIT ($33)

b. SST

EXTSST (0)
SLOCK (1)
BATT (10)
CATT (11)
CFFA (12)
IBL (14)
TIQ (15)
CLOCKS (17)
ATTLNG (19)
STEMP (20)

c. EST

LDDA (9)
DSKST (29)

d. ATT

FTOB (1)
JIDK (9)
FCNT (3)
RCNT (4)
TRB (6)
PFBA (7)
CURDA (9)
FMTNO (13)
RFA (14)
BMODE (15)
ASCNT (5)
CMODE (2)
ELDT (16)
DOCNO (17)
VR (29)
SW1 (19)
SW2 (20)
FUNIDK (21)
AINT (27)
ASRCH (28)
ATOPT (33)
LUN (35)
TCLK (36)
ATEMP (73)
SC (36+10)

AA5641

TJOB (SC)	Job name
TBATCH (SC+3)	Batch number
TATT (SC+4)	Station number
TEMP (SC+5)	
ABTINF (SC+6)	first word of ABT entry
TOPR (SC+7)	operator number
RCD (32)	
RCDER (30)	
VRCC (16)	
KEYDPR (26)	
STS (31)	the beginning of record for the statistic file
SBATCH (1)	
SOPR (2)	statistic
START (3)	
SDAY (4)	
ACC (5)	
SKEY (6)	record
SVRCC (7)	
SRCDER (8)	
SRCD (9)	
SYEAR (10)	
SJOB (13)	

e. TRANSFER VECTOR

CLCDWA ($90)
ENTSR ($91)
EXTSR ($92)
GETBIN ($99)
GETALP ($9B)
GFCORE ($9D)
DISP ($A0)
SPILL ($A1)
READN ($A6)
REDDWA ($A9)
WRTDWA ($AA)
RELFOR ($AD)
CAM ($AE)
WAIT ($B0)
SV ($B1)
REJSUP ($B2)

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __THW - Thaw__   Level 6

2. MACROS USED

    None.

3. PROGRAM DEPENDENCIES

   a. ENTRY POINTS

      One entry point THW

   b. EXTERNAL LIST, ROUTINES CALLED VIA TRANSFER VECTOR

ENEKSR	ENTSR EXTSR SETMOV WAIT
PIO	WSS CLCDWA READN REDDWA WRTDWA
GFCORE	GFCORE RELFOR
CAM	CAM
SUPER	SV REJSUP
GETLJD	
SPILL	SPILL
DISP	REGEN SETDIS
SCANV	GETBIN GETALP
ARITH	BD
VMM	EORTST VMM1
EMM	STTYPE

    SYSTEM EXTERNALS

          AMONTO, ADAYTO, AYERTO - The date used for the statistic's record.

AA5641   PRINTED IN THE USA

4. DATA AREAS AND SYSTEM TABLES USED

    LJD
    ABT
    SST
    ATT
    TOPT
    FTOB - ATT image and statistic information in the first sector

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    Program size $271_{16} = 625_{10}$.

    TRB according to job maximum record length.

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

    Partially locked. If the "batch status modification" control bit is set
    (bit 13 in SST+1) the program's execution is delayed until the lock is
    cleared.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___TOF - Turn Off Key Entry Station_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    TOF is entered by the supervisor when one or more key entry stations
    have to be locked. The station is not powered down.

2. DETAILED DESCRIPTION

    TOF starts by making itself movable and checking that TOF is filled by
    a comma. If not, an error message is sent to the supervisor and the
    command is rejected. Next, the first terminal number is read and stored
    in ATT/TNO. If followed by a comma, a second number (last terminal) is
    read and stored in ATT/TNOMAX. Errors occur if:

    a. First number is illegal or zero;

    b. Second number is illegal or zero or nonexistent (after comma);

    c. Second number is less than the first one;

    d. The first number (or the second, when specified) is greater than
       the actual number of active terminals in the system.

    e. Comma or other character follows the second number.

    If no such errors detected, the program sets a flag (ATT/FLAG) to 0 if
    only one station is specified, or to 1 when more stations have to be
    locked. Also, ATT/FLAG1 contains -0 as an indicator to print the Reader

        TOF DONE AT THE FOLLOWING TERMINALS:

    The program enters a loop (headed GOOD) during which:

    ● Address of the ATT corresponding to the actual terminal number (TNO)
      is calculated.

    ● First word of the ATT is checked. If that is a supervisor terminal
      (bit 7 = 1) it is skipped. Otherwise, bit 15 of that word is set,
      locking that terminal.

    ● The terminal number is stored in NUMB and transferred to the interrogate
      buffer (output line). When FLAG1 = 12, that line is printed out,
      FLAG 1 is set to 0 and subsequent terminal numbers are entered from the
      beginning of the line.

    ● TNØ is increased and checked against TNØMAX. TNØ greater than
      TNØMAX indicates the end of looping. FLAG1 is set to 0 and the
      last line is printed. One more checking of FLAG1 results in
      returning control to SV.

      TNØ less or equal to TNØMAX causes TNØ to be stored in ASCII form
      on the output line and FLAG1 to be increased, and the loop is
      entered again.

AA5641

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___TOF - Turn Off Key Entry Station___ Level 6

INTERFACE

1. ENTRY METHOD

   From DSKSUP, via ENTPRG.

2. INPUT PARAMETERS

   I = ATT base address.

3. EXIT METHOD

   JMP- (SV) is no errors. Else,
   JMP- (REJSUP)

4. OUTPUT PARAMETERS

   ATT word 1 bit 15 set (terminal locked) for the specified terminals.

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      SST ($47)
      ONEBIT ($23)
      GETALF ($90)
      SUPRW ($97)
      SV ($B1)
      LPMASK ($2)
      GETBIN ($99)
      TEN ($46)
      REJSUP ($B2)
      ZERO ($22)

   b. ATT

      SLOCK (1)
      ATTLNG (19)
      FLAG (46)
      TNOMAX (48)
      NUMB (50)
      ATTB (10)
      FW (3)
      TNO (47)
      FLAG1 (51)
      AINT (27)

   c. CONSTANTS

      ($46) = 10

   d. EXTERNALS

      SETMOV
      PRTMST

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   GETBIN
   SUPRW
   SV
   SETMOV
   SETALP
   REJSUP
   PRTMSG

4. DATA AREAS USED

   SST
   ATT
   Interrogate buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENT

   $68_{16} = 104_{10}$ words.

2. RELOCATION TYPE

   Loader relocatable, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant.

AA5641

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ TON - Turn On Key Entry Station   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    TON is entered by the supervisor when one or more terminals have to
    be reactivated. After that command, operator can continue entering
    data, commands, etc.

2. DETAILED DESCRIPTION

    Refer to the TOF program documentation for description of operation,
    data areas used, equates, labels, etc. The two programs are
    identical in everything except that instruction 35 of TOF:
    EOR - ONEBIT + 15, is missing in TON. The result is that bit 15
    of word 1 in ATT is cleared, thus reactivating the corresponding
    terminal.

DOCUMENT CLASS _____TOTAL_____

PROGRAM NAME_____Display A Specified Counter Value_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    There are two entry points in TOTAL: TOT and TOT2. TOT is entered when a
    TOT, n command is keyed: the contents of batch counter n are displayed. TOT2
    is entered from the End of Batch processor (EOB) when an unbalanced batch
    counter is detected. The batch counter is displayed.

2. DETAILED DESCRIPTION

    a. TOT

        1. (TOT). Set ATT/ATEMP + 2 = 3 (Entry from TOT, n command).

        2. Set ATT/SW/7,8 (7: Reset or Disp must be next character, 8: Reset
            must regenerate display).

        3. Call GETALP to get the next input character. If next character is
            not a comma, set (Q) = $12 and exit to ERROR to display error code
            E1 (ILLEGAL REQUEST).

        4. (GETCTR). Call GETBIN to get counter number from input string. If
            an error is detected, or if the counter number is terminated by a
            comma, go to step 6.

        5. (CHKCTR). If 0 ≤ counter number ≠ 7, go to step g.

        6. (TOERR). Set (Q) = $13 and exit to ERROR to display error code E2
            (ILLEGAL PARAMETER).

        7. (OK). Call GETBUF to allocate a 28 word buffer. Call DISPLY to clean
            the display. Set up the message 'COUNTER N = ' in words 4-9 of the
            buffer block.

        8. Call UNP to convert counter value to ASCII digits. Move ASCII value
            to buffer, substituting blanks for pairs of leading zeros.

        9. (EOL). Call DISPLY to display the message: COUNTER N = xxxxxxx

        10. If ATT/ATEMP + 2 < 0 (called from EOB) set Q = $1C and exit to ERROR
            to display error code C6 (BALANCING ERROR).

        11. (TOTEXT). Set ATT/2/14 = 1 (IDK mode) and call CLRSR to exit.

    b. TOT2

        1. (TOT2). Set ATT/ATEMP + 2 = -0 (Entry from EOB).

        2. Get counter number in A register and go to step 1.2.

64

DOCUMENT CLASS __TOTAL__

PROGRAM NAME__ Display A Specified Counter Value__

INTERFACE

1. ENTRY METHOD

   TOTAL is a disk resident program which is called via ENTPRG (Enter a disk resident program).

2. INPUT PARAMETERS OR CONDITIONS

   (I) = Base address of Active Terminal Table (ATT)

3. EXIT METHOD

   If entry was due to TOT, n command, exit is to dispatcher via CLRSR. If entry was from EOB, exit is to ERROR. If an error is detected in input data, exit is to ERROR.

4. OUTPUT PARAMETERS OR CONDITIONS

   (I) = Base address of ATT

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      ONEBIT ($23)
      ZROBIT ($33)
      SST ($47)
      CLRSR ($93)
      ERROR ($96)
      GETBIN ($99)
      GETALP ($9B)
      DISPLY ($Ao)

   b. SST

      STEMP (20)
      STEMP2 (21)

   c. ATT

      SW1 (19)
      AINT (46)
      ATEMP (73)

2. MACROS USED

   None

65

DOCUMENT CLASS __TOTAL__

PROGRAM NAME __Display A Specified Counter Value__

3. PROGRAM DEPENDENCIES

    CLRSR   Clear exits stack and function lock
    ERROR   Display error
    GETBIN Get a number from input string; convert it to binary
    GETALP Get an alpha character from input string
    DISPLY Display a message

4. DATA AREAS USED

    SST   System Status Table
    ATT   Active Terminal Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $6A_{16} = 106_{10}$ words

2. RELOCATION TYPE

    TOTAL is a disk resident program.
    It is coded in run anywhere form.

3. ENTRANCY CLASSIFICATION

    Reentrant

66

DOCUMENT CLASS __CYBERDATA_____

PROGRAM NAME___ __TRC - Tape Recovery and Leader Functions_____ __Level 06__

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    The program processes the two commands, HDR and TRC.

    a. HDR

        The HDR command writes a special CYBERDATA system label on tape which
        is checked by the WBT command - If this label is not found, the WBT
        command is rejected -

        HDR $\left\{ \,jjjjjj \ (1) \right\}$ $\left\{ ,mmmmm \ (2) \right\}$
                                                        $\underline{0}$

    where

        (1) jobname, if missing this tape will be dedicated to system
             use, i.e. Dump.

        (2) number of blocks to be skipped before writing CYBERDATA label.
             (Default is 0).

    b. TRC

        The TRC command is used to recover data on magnetic tape if a failure
        occurred while batches were being recorded.

        The command format is as follows:

            TRC, mm (1)

    where

            i. recovery area number from last wbt message.
            The program has 2 entry points: TRC, ENT2.

2. DETAILED DESCRIPTION

    (First entry point)

    Program calls SETMOV to clear. I/O bit in Header. Next it rewinds the
    tape. In case of tape I/O error, return is made to supervisor mode.
    A check is made to ascertain which command is to be processed.

a. HDR COMMAND PROCESSING

A syntactic check is made - In case of error return is mode to
supervisor mode.

In case the job name parameter was present GETLJD Is invoked
to get the corresponding entry in LJD. In case of Invalid job name
return is made to supervisor mode. Mag Tape format number is extracted
from LJD entry.

A buffer for preparing the CYBERDATA System label (EOD) is requested
by use of CAM. In case of no core available (after 100 cycles) return
is made to REJSUP.

After fixing the backward pointer the buffer is cleared.

In case the job name was present (not for system use) the corresponding
Mag Tape format is used Into core (by use of REDDWA). In case the
format is inactive return is made (via MATER).

The recording mode is checked. In case the receive mode In MTF was oo
(default) the actual mode Is fixed. (In case of invalid receive mode
return is made (via MATER). (Note that for system use it has no meaning.)

Advance is made on the tape according to the number of blocks to be
skipped before writing the label (entered in the command) - Each time
a block is advanced a check is made if CANCEL was entered.

Each block read is checked for Tape Mark or End of Tape. In case of End
of Tape return is made via MATER (after printing error message).

It is found (before the requested number of blocks were advanced) return
is made via MATER (after printing error message).

Otherwise after skipping the requested number of blocks, the EOD (still in
buffer) is updated - A Tape Mark is written on tape followed by a Leader
of 16 words containing "EOD1" in the first two words and "*" or blanks
in the following words. (For system use blanks are filled in.)   - Next
the buffer containing the EOD is written on tape. (In case of Tape I/O
error return is made via MATER.)

After writing the EOD2 a short block is written (noise block) - It is done
to prevent the search of the EOD1 block all over the tape In case a wrong
tape (recorded in the different code) is mounted. The reading of the noise
block will cause a short read condition when trying to find the EOD1.

b. TRC COMMAND PROCESSING

Syntactic check is made on the command. In case of format error or error
in the number of receiving area, return is made via REJSUP.

Next a buffer is requested (as in HDR).

Control word of recovery area read into core and validity of recovery
area number is checked. In case of illegal number and return is made
via REJSUP (second entry point).

EOD is read from the recovery area into core. In case the number of blocks
in reel is less, program returns with error, otherwise it enters the
segment described in HDR to read the MTF, get the recording mode and
advance the tape number of block indicated - the third word in EOD.
Mode of blocking is checked - In case it is fixed blocking and the
last write on tape did not fill the block, (it means the current batch
occupies part of that block).

The block is read into core (according to the recording mode), filled
with the padding character (according to specification in first word of
MTF), and written again on tape.

A message announcing the last recorded batch is printed.

Program exits to MATER if file number is zero, otherwise it exits
LBL0 or LBL1 according to file label type.

INTERFACE

1. ENTRY METHOD

TRC is a disk resident program invoked by ENTPRG.
It has two entry points.

The program is entered via the first entry point when invoked (by ENTPRG)
from DSKSUP (processing of the supervisor commands TRC and HDR).

The program is entered via the second entry point when called from MTFMON.

2. INPUT PARAMETERS

When entered via the first entry point, parameters are included in the command itself (see command format in general description).

When entered via the second entry point the following parameters are passed.

```
MTFENT:   20
MTFEXT:   MATER INDEX
TRALNG:   EOD1 length
TRANUM:   Recovery Area number
MFT:      First word of Mag Tape format
OUTBUF:   pointer to buffer
```

3. OUTPUT PARAMETERS

JIDX - 0 if HDR command was used to open tape dedicated to system use. Job name index in LJD otherwise.

INBUF - 0

TRB - 0

OUTBUF - pointer to output buffer

EODADR - pointer to buffer for EOD

MTFEXT - index of MATER

MTFENT - 7FFF if exit is to LBL0 or LBL1
         0   if exit is to SV
         S   if exit is to MATER

4. EXIT METHOD

a. JMP - (SV)
   normal completion of HDR processing

b. JMP - (REJSUP)
   in case of:

   1)   Illegal recovery area

   2)   Invalid recording area

   3)   Tape I/O error (when rewinding)

   4)   Illegal parameters

    c. To error entry in MATER

        ENA 3
        LDQ MTFEXT,I
        RTJ - (ENTPRG)
        in case of tape I/O error when working

    d. To MATER (first entry point)

        LDQ MTFEXT,I
        RTJ - (ENTPRG)
        in case file # = 0

    e. To LBL0 or LBL1 (third entry) .
        in case file # = 0

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. TRANSFER VECTOR

        REDDWA ($A9)
        CLCDWA ($90)
        TAPIO ($9A)
        SUPRW ($97)
        RELFOR ($AD)
        CAM ($AE)
        WAIT ($B0)
        SV ($B1)
        REJSUP ($B2)
        EXTSR ($92)
        ENTSR ($91)
        EXDFUN ($9F)
        ENTPRG ($9E)
        RTNSR ($98)
        GETBIN ($99)
        GETALP ($9B)
        GETCOM ($9C)

b. ATT,SST,EST

MTFDA (42)
STEMP (20)
SST ($47)
ATEMP (73)
JIDX (3)
INBUF (5)
TRB (6)
OUTBUF (28)
IRB (27)
SVCA (36)
MTCA (46)
TRMNT (23)
MTFENT (MTCA)
MTFEXT (MTCA+1)
EODADR (MTCA+2)
MTF (MTCA+3)
BCOUNT (SVCA+5)
TRANUM (MTCA+11)
POSITN (MTCA+12)
TM (MTCA+13)
MAXTRA (32)
TRALNG (40)

c. EOD1 EQUATES

TOT (2)
MTFN (3)
FILE (4)
JOB (5)
BATCH (8)
LBC (12)

d. GENERAL

LPMASK (2)
NZERO ($12)
ZERO ($22)
ONEBIT ($23)
ZROBIT ($33)
MO (4)
BFR (10)
BFW (11)
CFR (18)
CFW (19)
FR (2)
FW (3)
MATER ($0305)

2. MACROS USED

None.

3. PROGRAM DEPENDENCIES

SETMOV
MOVREC
BD
GETLJD
PRTMSG
REDDWA
CLCDWA
TAPIO
SUPRW
RELFOR
CAM
WAIT
SV
REJSUP
EXTSR
ENTSR
RTNSR
EXDFUN
ENTPRG
GETBIN
GETALP
GETCOM

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___TRC - Tape Recovery and Leading Functions_____ Level 06

4. AREAS AND SYSTEM TABLES

    SST
    EST
    ATT
    MTCA
    EOD
    MTF
    LJD
    Tape Recovery area
    IRB

5. INTERNAL SUBROUTINES

    GETCOR - To get a buffer
             ON ENTRY - Q size of buffer
             ON EXIT - Q pointer to buffer
    in case no one is available it exits error entry in MATER

    REDWRD - To read into core from disk
             ON ENTRY - A - sector address
                        Q - displacement
                        NOWRD - number of words to be read
                        CORAD - pointer to buffer in core
    IOON   - To change the I/O bit of a buffer
             ON EXIT - Q points to beginning of data
    IOOF   - To clear the I/O bit of a buffer
             ON EXIT - Q points to beginning of data

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    267_{16} = 615 words of program
    +43 words of buffer for EOD
    +buffer for a block

2. RELOCATION TYPE

    Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME___ TST - Diagnostic Software for Remote Cluster _____ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    TST is a supervisor function executed by the central supervisor together
    with the Customer Engineer.

    The format of the command is as follows:

        TST,nn(1),n(2)

    where: nn(1) = equipment number (01-15)
             n(2) = number of sations to be checked (1-8)

    After checking the parameters 'READY?' is printed on the list unit waiting
    for the Customer Engineer to loop back the suspect device for the supervisor
    to enter Y,N or C;

    where: N = No further checking. Stop the test.
             Y = Yes, perform the test for the specified time.
             C = Continuously, perform the test continuously.

2. DETAILED DESCRIPTION

    TST, after allowing itself to be moved (SETMOV) is looking for its parameters.
    First, it checks whether a comma is following the TST command. If comma was
    not found a jump is made to REJSUP indicating 'INVALID COMMAND FORMAT',
    otherwise the program proceeds by looking for the equipment number and the
    number of stations to be tested. The equipment number given is checked against
    the equipment numbers saved in the EST (WES). If the number given is illegal
    exit is made to REJSUP indicating "INVALID PARAMETER". The stations specified
    by the 'number of stations' parameter are checked to be declared in CYBERDATA
    start-up time (SLOCK in SST) and to be in IDLE mode. In case a station is
    either function locked (LOCK,B14) or pseudo function locked (LOCK,B13) on in
    Interrogate mode (CMODE,B15), the code of 'CANCEL' key is put in the TIQ
    and a delay is taken waiting for the completion of the function performed
    on the station. The program clears the error lock (LOCK,B11) and the 'RESET
    must be next' (SW1,B7,B8) and sets the function lock in order to skip over
    the normal character processing via MPC. The program waits also for the
    completion of the output in case it is threaded for display.

TST function prints READY? via PRTMSG and reads a character followed by
a CR via SUPRW. The character entered is checked to be N,Y or C; in case
neither of them were entered, 'READY?' is printed again waiting for a
new character to be entered.

If 'N' was entered, the stations are released from being 'function locked'
and an exit is made to SV.

In case 'Y' was entered, the cycle rate for testing is set to 3, while if
'C' was entered the cycle rate is set to 50. The TIQ is cleared out for
each station and initial values are given for:

- error count (ERROTB)
- cycle rate (FIFTY)
- character to be sent (CHARTB)
- delay for reply (TENTB)

TST, after setting up the parameters is called by MPC to perform the
Input/Output for the stations. When a character is sent to the station
the 'wait for reply' counter is set to 1. In every interrupt cycle there
is a check whether the character was returned.

Errors are counted in the following situations:

a. External or Internal reject when sending a character.

b. No reply during 10 cycles of interrupts.

c. The interrupt character is different from the output one. (This indication
    will also be received when an error is signified in the input status
    word. In this case, VLINP sets the character for an invalid code).

When all the characters 01 through 7F were sent to all the stations the
program checks whether the 'error count' is to be printed on the list unit.
The message will be printed every 3 cycles if 'Y' was entered, or 50 cycles
if 'C' was entered (a cycle here means the set of codes between 01 and 7F).
If 'Y' was entered a header is printed followed by the error count and the
program loops back to the point where 'READY?' was printed.

If 'C' was entered, the header will be printed only the first time and then
it will be followed by the 'error count'. The program loops back to the
point where initial values are given to the initial flags.

Every interrupt cycle and every message print, a check is made whether the
function was asked to be terminated by MI SX. In this case, the error count
to this point is printed and a jump is made to REJSUP indicating "FUNCTION
TERMINATED".

INTERFACE

1. ENTRY METHOD

    TST is a central supervisor function entered by ENTPRG.

2. INPUT PARAMETERS OR CONDITIONS

    (See General Description above.)

3. EXIT METHOD

    Normal exit - jump to SV.
    Error exit or Function termination - jump to REJSUP.

4. OUTPUT PARAMETERS OR CONDITIONS

    Error log for all the specified stations.
    I = current ATT base address.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        SST ($47)
        LPMASK ($2)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SUPRW ($97)
        GETBIN ($99)
        GETALP ($9B)
        DISP ($A0)
        WAIT ($B0)
        SV ($B1)
        REJSUP ($B2)

b. SST

    SLOCK (1)
    ATTB (10)
    IBL (14)
    TIQ (15)
    ATTLNG (19)
    STEMP (20)

c. EST

    WES (30)

d. ATT

    CMODE (2)
    SW1 (19)
    SW2 (20)
    CHKSX (23)
    AINT (27)
    TOPT (33)
    EQU in ATT for temporary work area of ATT 0
    CYCLE (50)
    PFLAG (53)
    NEWTER (56)
    COUNT (46)
    EQUNO (47)
    TERNO (48)
    NOOFST (49)
    LASTRN (51)
    LOPCUN (52)
    SAVEQ (54)
    EQU in ATT for temporary work area for the stations to be checked.
    FIFTY (50)
    CHARTB (53)
    TENTB (54)
    ERROTB (55)

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ TST - Diagnostic Software for Remote Cluster ___ Level 06

e. CONSTANTS

FR (2)	Format Read code
FW (3)	Format Write code
CYCLEY (3)	No. of loop cycles when Y is entered.
CYCLEC (50)	No. of loop cycles when C is entered.
Y ($59)	ASCII of Y
N ($4E)	ASCII of N
C ($43)	ASCII of C

2. EXTERNALS

SETMOV
PRTMSG
BD
CANCEL   Entry point in VLINP for CANCEL code.

3. PROGRAM DEPENDENCIES

PIO	(SUPRW)	Read/Write to supervisor devices.
SCANV	(GETBIN)	Get binary number from input string.
	(GETALP)	Get alpha-numeric character from input string.
SUPER	(SV)	Normal exit from supervisor function.
	(REJSUP)	Error exit from supervisor function.
	PRTMSG	Print the identified message from the REJMSG buffer.
ARITH	(BD)	Binary to decimal conversion.
ENEXSR	(WAIT)	Wait one interrupt cycle.
	(SETMOV)	Set the program to be movable.
VLINP	(CANCEL)	ASCII code for CANCEL key.

4. DATA AREAS USED

Interrogate buffer
ATT
TIQ

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___ TST - Diagnostic Software for Remote Cluster _____ Level 06

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   16 = 10 words.

2. RELOCATION TYPE

   Run anywhere and movable.

3. ENTRANCY CLASSIFICATION

   Reentrant.

DOCUMENT CLASS ___VERIFY___

PROGRAM NAME___Process VERIFY Command___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    VERIFY interprets the VER, Job name, batch number, Operator number statement, sets up the Active Batch Table and Active Terminal Table to VERIFY status, and positions to the beginning of the batch.

2. DETAILED DESCRIPTION

    a. (VER). Call SETMOV to enable moving of VERIFY program area.

    • If ATT/CMODE/8-11 = 0 (Idle mode), go to Step b.

    • If not, set ATT/SW1/8 = 1 (Reset must regenerate display), and exit to ERROR with (Q) = CE1.
      (ILLEGAL REQUEST error.)

    b. (CONT). Call GETALP to get the 4th character in the input string.

    • If it is not a comma, go to Step d.

    • Set ATT/TJOB through ATT/TJOB + 2 (Job name area) to spaces.

    c. (LOOP). Call GETALP to get a character of Job name.

    • If character is a valid ASCII character, go to Step f.

    d. (ERR). Set (Q) = CD4 (INVALID JOB NAME error).

    e. (ERR1). Set ATT/SW1 = 0100_{16} (reset must regenerate display).
       Set ATT/VMOD/12-15 = 0 (Basic mode = idle). Exit to ERROR.

    f. (NXTCHR). If the input character is a comma, go to Step g.

    • Save Input character in next available character position in ATT/TJOB through ATT/TJOB + 2.

    • If input Job name is more than 6 characters long, go to Step d.

    • If not, repeat from Step c.

    g. (GETBCH). Call GETBIN to get batch number from input string and convert it to a binary number.

    • If the characters entered for the batch number are valid ones, i.e. numeric characters which were terminated by a comma and the number does not exceed 999, go to Step b.

    g1. Set (Q) = CD5 (INVALID BATCH NUMBER error) and go to Step e.

DOCUMENT CLASS ___VERIFY___

PROGRAM NAME_____Process VERIFY Command_____

h. (GETORR). Call GETBIN to get operator number from input string and convert it to a binary number.

- If a nonnumeric character is in the input string, or if input is not terminated with EOR, or if operator number exceeds 999, set (Q) = CD1 (INVALID OPERATOR NO. error), and go to Step e.

- Release interrogate buffer.

- Clear ATT/3 through ATT/32.

- Set ATT/ECODE = 0.

- Call BD to convert batch number to packed decimal, and save it in ATT/BATCH.

- Call BD to convert operator number to packed decimal, and save it in ATT/TOPR

i. (RTRY). If ATT/0/10 = 1 (CRT terminal), save job name in TOPT words 5, 6 and 7.

- Call GETLJD to locate the designated job in the Legal Job Directory.

- If GETLJD program is locked, repeat Step i.

- If designated job does not exist in LJD, go to Step d.

- Save job index in ATT/JIDX.

- Call SPILL to assign an entry in the Active Batch Table.

- If designated batch is not found in ABT, or if batch is not waiting for verification, go to Step g1.

- Save address of first track in ATT/FTOB.

- Save batch number in ATT/BATCH.

- If disk assigned to the batch is down, set (Q) = CF6 (DISK DOWN error), and go to Step e.

j. (VERI). Call CAM to allocate a Terminal Record Buffer.

- If space is not available, set (Q) = CF4 (CORE FULL, TRY AGAIN error) and go to Step e.

- Set up linkage to TRB.

- Call SPILL to set batch to Verify mode in ABT.

- Set ATT/CMODE = 0100_{16} (Verify mode).

- Set ATT/JIDX/8-15 = Job index.

364

DOCUMENT CLASS ___VERIFY_____

PROGRAM NAME_____Process VERIFY Command_____

- Set ATT/BMOD = 1000_{16} (Basic Mode = Verify).
- Set ATT/SW2/2:0 according to batch verification type.
- Set ATT/SW2/3 = 1 (Monitor ATT check override).
- Set ATT/TCLK = Time in seconds since start-up.
- Set ATT/TSBCH = Batch start-up time in minutes.
- Call CLCDWA to calculate the disk word address of word 91 of the batch.
- Call WRTDWA to write Operator number, Batch start-up time, Run start-up time, Accumulated time (=0) and record count from beginning of batch (=0) in words 91-95 of the first sector of the batch.
- Call SUBR to restore ATT/8 through ATT/12 from second sector of batch.
- Call SUBR to restore ATT/29 from second sector of batch.
- Call SUBR to restore ATT/36 through ATT/63 from second sector of batch.
- Call CLCDWA to calculate disk word address of first physical record of batch.
- Save address of previous record in Terminal Record Buffer.

k. (BACK). Call READP to read the record.

- Repeat Step k until first logical record of batch is found.
- Call GFCORE to read applicable format description into core.
- Call DISP to set screen location to data area.
- Call WRTATT to write ATT to disk.
- Call EXDFUN to release VERIFY program area.
- Exit to EORVRC.

INTERFACE

1. ENTRY METHOD

   VERIFY is a disk-resident program which is called via the ENTPRG program.

2. INPUT PARAMETERS OR CONDITIONS

   (I) = Base address of Active Terminal Table (ATT).

3. EXIT METHOD

   Normal exit is to EORVRC.
   Abnormal exit is to ERROR.

365

DOCUMENT CLASS ___VERIFY___

PROGRAM NAME___Process VERIFY Command___

4. OUTPUT PARAMETERS OR CONDITIONS

(1) = Base address of ATT.
ATT is set to verify the designated batch.

ASSEMBLY CHARACTERISTICS

1. EQUATES

a. LOCORE

LPMASK (2)
NZERO ($12)
ZERO ($22)
ONEBIT ($23)
ZROBIT ($33)
SST ($47)
CLCDWA ($90)
ENTSR ($91)
EXTSR ($92)
ERROR ($96)
GETBIN ($99)
GETALP ($9B)
GFCORE ($9D)
EXDFUN ($9F)
DISP ($A0)
SPILL ($A1)
READP ($A5)
REDDWA ($A9)
WRTDWA ($AA)
WRTATT ($AC)
CAM ($AE)
WAIT ($B0)

b. SST

EXTSST (0)
CLOCKM (16)
CLOCKS (17)
STEMP (20)

c. EST

DSKST (29)

366

DOCUMENT CLASS ___VERIFY___

PROGRAM NAME___Process VERIFY Command___

    d. ATT

        FTOB (1)
        CMODE (2)
        JIDX (3)
        TRBA (6)
        TOPR (8)
        TSBCH (9)   Operator number (saved temporarily)
        TCLK (10)   Batch number   (saved temporarily)
        BMOD (15)   Clock value   (saved temporarily)
        SW1 (19)
        SW2 (20)
        FUNIDX (21)
        ABTIDX (22)
        SWITCH (24)
        AINT (27)
        ATOPT (33)
        ECODE (34)
        BATCH (35)
        TJOB (36   Job name   (saved temporarily)
        ATEMP (73)
        ATEMP1 (74)
        ATEMP2 (75)
        ATEMP3 (76)

    e. ERROR CODES

        CD1 ($D)
        CD4 ($10)
        CD5 ($11)
        CE1 ($12)
        CF4 ($17)
        CF6 ($1E)

    f. EXTERNALS

        SETMOV
        DB
        GETLJD
        SETDIS   Parameter for DISPLY to set screen location according to (A).
        EORVRC

2. MACROS USED

None.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___VERIFY___

PROGRAM NAME___ Process VERIFY Command___

3. PROGRAM DEPENDENCIES

	ENTSR	Enter Subroutine
ENEXSR	EXTSR	Exit Subroutine
	WAIT	
	SETMOV	Enable Move of Program Area
ENTPRG	EXDFUN	Release the disk resident program.
GSE	ERROR	Display error at terminal and terminate the function
	CLCDWA	Calculate Disk Word Address.
	READP	Read Record according to previous disk address saved in TRB.
PIO	REDDWA	Read Disk Word Address
	WRTATT	Write ATT
SCANV	GETBIN	Convert Input Number to Binary.
	GETALP	Get Alpha Character from Input String.
GFCORE		Get Format into Core
DISPLY		Set screen location to data.
SPILL		Maintain Active Batch Table.
CAM		Get core for TRB.
GETLJD		Locate Entry in Legal Job Directory.
EORVR	EORVRC	End of Record Processor-Verify Mode.

4. DATA AREAS USED

SST	System Status Table
ATT	Active Terminal Table
TRB	Terminal Record Buffer
TOPT	Terminal Output Table
INT	Interrogate Buffer
DTL	Data Track Layout

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$113_{16} = 275_{10}$

2. RELOCATION TYPE

Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

Reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___VLBGOP - Background Core and Operation Mode Processor   Level 06___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

VLBGOP is an overlay called by the manual interrupt processor MIPRO when
the operator enters MI VLBG. The program outputs "VLBG IN" to the system
comment device and the operator must type in two parameters CORE, MODE.
The CORE parameter must be a four-digit hexadecimal number, and the MODE
parameter must be either: "MUST", meaning background core must be opera-
tional; "NONE", meaning no background is requested; or "RQST", where
background is requested but will be operational only if possible. The
minimum background core requirement is $0400. Any change made by VLBGOP
will take place at the next startup.

2. DETAILED DESCRIPTION

VLBGOP first stores its load address for later use in release. The input
buffer is initialized to all ones and "VLBG IN" is output to the system
comment device. Via a FREAD an input request is made to the system control
device and VLBGOP jumps to the dispatcher to wait until the operator types
in parameters. When the input is complete, error checks are made. If an
I/O error occurs, VLBGOP outputs the statement "I/O ERROR" to the system
comment device and starts again by outputting "VLBG IN". If the core value
entered was not a four-digit hexadecimal number greater than or equal to
$0400, or if the mode was not "MUST", "NONE", or "RQST", the statement
"DATA ERROR" is output to the system comment device and the process begins
again at "VLBG IN". When the parameters are entered correctly, the core
image on disk is rewritten with background core value and background mode
flag (MUST = 1, RQST = 0, NONE = 1) updated. The statement "VLBG OUT" is
output to the system comment device and VLBGOP releases its own memory
space and jumps to the dispatcher.

INTERFACE

1. ENTRY METHOD

Execution begins at the first instruction of this system loaded overlay.

2. INPUT PARAMETERS OR CONDITIONS

Operator enters MI VLBG.
No parameters are initially passed to the overlay.
The memory address where VLBGOP is loaded is in the location immediately
preceding the first instruction.

3. EXIT METHOD

Jump to the dispatcher.

4. OUTPUT PARAMETERS OR CONDITIONS

In the core image on disk:

Background core requirement is set.

Background mode is set to:

    1 = MUST have background operational
    0 = RQST have background operational
   -1 = NONE background

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        DISPCH ($EA)

    b. CONSTANT

        MINBG ($0400)   Minimum amount of background

2. MACROS USED

    FWRITE
    FREAD
    RELEAS

3. PROGRAM DEPENDENCIES

    BGCORE is an external for amount of background (contained in SYSDAT)

4. DATA AREAS USED

    Internal buffer

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $AA_{16} = 170_{10}$ words.

2. RELOCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Not reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME__ VLINP - Terminal Keyboard Driver __ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    VLINP does an input from each terminal that is not supervisor or system
    locked. If a character was ready, the character is stored in the terminal
    input queue. One input from each unlocked terminal is performed each time
    VLINP is called by VLINT. VLINP runs at the same priority level as VLINT
    (currently 10).

2. DETAILED DESCRIPTION

    The maximum number of characters in a TIQ is calculated and stored in
    program data area. If the system is locked, VLINP exits immediately.
    Otherwise, an overflow counter is created from the number of terminals
    in the system and the input loop begins. Each terminal is checked for
    supervisor or system lock. If locked, VLINP goes on the the next terminal.
    If the terminal is not locked, an input is performed. If the input is
    rejected, VLINP goes on to the next terminal. If the input occurs, bit 11
    is checked for data ready. If no data is ready, VLINP goes on to the next
    terminal. If a character is ready, it is checked for the pattern when the
    PDU is off, if the code is equal, VLINP goes to the next terminal; else bits
    15, 9, 8 are checked for parity, character lost, and line break errors. If
    an error is found, the character count in TIQ is incremented, but the
    character is zeroed out. For any character input, the TIQ character count
    is incremented. If the count reaches the maximum TIQ count, ATT/SLOCK/12
    and ATT/SW2/8 are set to indicate that the TIQ is full and "LL" should be
    displayed. The character is stored in the next character position of the
    TIQ. If the character input is a cancel, ATT/FRMNT/15 is checked to see if
    current function can be terminated. If so, ATT/FRMNT/14 is set to indicate
    cancel was entered and processing continues with the next terminal. If not,
    the cancel is entered in the TIQ as any other character would be. When all
    the terminals have been processed, VLINP returns control to the caller.

INTERFACE

1. ENTRY METHOD

    Return jump

2. INPUT PARAMETERS OR CONDITIONS

    VLINP is called after each terminal controller interrupt.

3. EXIT METHOD

    Return to caller at P + 1.

4. OUTPUT PARAMETERS OR CONDITIONS

    Possible input of one character from each terminal.

AA5641

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __VLINP - Terminal Keyboard Driver__   Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE

      LPMASK (2)
      NZERO ($12)
      ONEBIT ($23)
      ZROBIT ($33)
      SST ($47)

   b. SST

      EXTSST (0)
      SLOCK (1)
      ATTB (10)
      IBL (14)
      TIQ (15)
      ATTLING (19)

   c. EST

      WES (30)

   d. ATT

      SW2 (20)
      FRMNT (23)
      TOPTAD (33)

   e. CONSTANTS

      ERRCD (0)
      CANCEL ($76)

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   None.

4. DATA AREAS USED

   ATT
   SST
   TIQ

AA5641   PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___VLINP - Terminal Keyboard Driver_____Level 6

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$71_{16} = 113_{10}$ words.

2. RELOCATION TYPE

Not reentrant.

AA5641

PRINTED IN THE USA

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    VLINT processes the periodic interrupt from the first terminal controller.
    If the operating system timer is present, VLINT schedules it. If Cyberdata
    is active, VLINT increments the clock, calls the terminal input driver,
    calls the terminal display driver(s), and schedules MPC (Main Process Control)
    if it is not still active. If Cyberdata is not active, exit is to the
    dispatcher.

2. DETAILED DESCRIPTION

    VLINT resets and enables the interrupt for the first (as listed in EST)
    terminal controller. If TIMINT has been installed in this system, VLINT
    schedules it (at priority 13). If Cyberdata is active, the interrupt
    counter is incremented and tested against the number of interrupts per
    second. When appropriate, the system seconds count is incremented. The
    terminal keyboard driver VLINP is called. VLINT examines the top of the
    thread (SST/THSLFS) for output to the 970-32's and calls VL2SCN if the
    thread is not empty. The 970-480 thread (SST/THCRT) is also checked and
    VL2CRT is called if the thread is not zero. SST/SLOCK/12 is checked for
    MPC busy. If MPC is not busy, VLINT schedules it at priority 5 with Q=0
    for terminal 0. Exit is to the dispatcher.

INTERFACE

1. ENTRY METHOD

    Direct jump.

2. INPUT PARAMETER

    Terminal controller interrupt occurred.

3. EXIT METHOD

    Jump to dispatcher.

4. OUTPUT PARAMETERS OR CONDITIONS

    Interrupt reset.
    Possibly TIMINT scheduled.
    MPC scheduled.

AA5641

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME__VLINT - Terminal Controller Interrupt Handler__ Level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        ONEBIT ($23)
        SST ($47)
        DISP ($EA)

    b. SST

        EXTSST (0)
        SLOCK (1)
        THCRT (4)
        THSLFS (5)
        CLOCKS (17)
        NTRVAL (18)

    c. EST

        WES (30)

    d. CONSTANTS

        CLRACT   Reset/enable controller interrupt code.
        LVL   Priority of TIMINT program.

    e. EXTERNALS

        TIMINT
        MIBVL   Flag indicating if CYBERDATA is working.
        CSRATE   Number of interrupts per second.
        VLINP
        MPC
        VLZCRT
        VLZSCN

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    TIMINT   The system's timer routine.
    SYSDAT (MIBVL,CSRATE)
    VLINP   Scan the channels for input.
    MPC   Schedule the main process control.
    VL2CRT   Send a display code to the CRT 970-480 stations.
    VL2SCN   Send a character to the self-scan 970-32 stations.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME____VLINT - Terminal Controller Interrupt Handler_____Level 6

4. DATA AREAS USED

　　SST
　　EST

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

　　$36_{16} = 54_{10}$　　words.

2. RELOCATION TYPE

　　Loader relocatable.

3. ENTRANCY CLASSIFICATION

　　Reentrant.

AA5641

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __VLOSOP__

PROGRAM NAME__Select Startup Conditions__

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

VLOSOP is an overlay called in by MIPRO when the operator enters MI VLOS.
The message "VLOS IN" is output to the standard system comment device. The
operator then enters; "STD," where the Cyberdata allocatable memory area is
a system constant; "NONE," where no area is reserved for Cyberdata; "BCKG,"
where Cyberdata allocatable memory is set to all available allocatable minus
an amount reserved for unprotected background programs. Any other entry
results in a "DATA ERROR" message. Then VLOSOP outputs "VLOS OUT" and
releases its memory space.

NOTE: The change in allocatable memory at startup is not effective until
the next autoload.

2. DETAILED DESCRIPTION

VLOSOP saves its address so that it can release its memory space when it
is finished. The input buffer is initialized with all bits set. The message
"VLOS IN" is output to the standard system comment device using a FWRITE
macro. A FREAD macro is used to put a two word input request against the
system control device. For any I/O errors the message "I/O ERROR" is out-
put to the system comment device. If the operator does not enter "STD,"
"NONE," or "BCKG," the message "DATA ERROR" is output to the system comment
device. If the operator inputs a correct parameter, the disk image of
Cyberdata active flag VLOS is set to 0 = STD, -1 = NONE, 1 = BCKG. The
message "VLOS OUT" is output to the system comment device and VLOSOP re-
leases the memory it occupied.

INTERFACE

1. ENTRY METHOD

Execution begins at the first instruction of the system loaded overlay.

2. INPUT PARAMETERS OR CONDITIONS

Operator hits MI and types VLOS.

No parameters are initially passed to the overlay.

The address where it is loaded is stored immediately preceding its first
instruction.

3. EXIT METHOD

Release core and go to dispatcher.

4. OUTPUT PARAMETERS OR CONDITIONS

Operator enters a valid VLOS parameter and exit is automatic.

DOCUMENT CLASS ___VLOSOP_____

PROGRAM NAME___Select Startup Conditions_____

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        DISPCH ($EA)

2. MACROS USED

    FWRITE
    FREAD
    RELEAS

3. PROGRAM DEPENDENCIES

    VLOS is an external for the Cyberdata active flag in SYSDAT.

4. DATA AREAS USED

    Internal buffer.

    VLOS is set in disk image.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $78_{16} = 120_{10}$ words
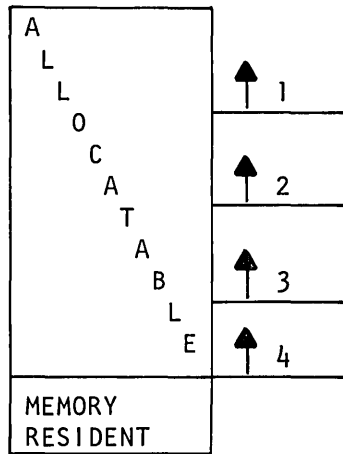
2. RELOCATION TYPE

    Run-anywhere

3. ENTRANCY CLASSIFICATION

    Not reentrant

ATTACH ANALYTICAL SUPPLEMENTS

Attachment A: Memory Allocation Writeup from
                Cyberdata Sofware User's Guide.

Attachment A

```
┌──────────────┐
│ A            │      ↑ 1     In the example at the left, only
│  L           │              Area 1 is available to programs
│   L          │              requesting space at priority 1.
│    O         │              Areas 1, 2, and 3 are available
│     C        │      ↑ 2     to programs requesting memory
│      A       │              at priority 3.  And all of allocat-
│       T      │              able memory is available to those
│        A     │              programs which request space at
│         B    │      ↑ 3     priority 4.
│          L   │
│           E  │      ↑ 4
├──────────────┤
│ MEMORY       │
│ RESIDENT     │
└──────────────┘
```

Example of Allocatable Memory

At startup, the CYBERDATA Operating System requests one large partition
of allocatable memory for table, data buffer, and disk resident program
use. This request is made at priority level 5, so the length of Area 5
determines what area CYBERDATA will always be able to receive, even if
lower priority programs are running.

The following parameters in SYSDAT determine the length of Area 5 and
the way that allocatable memory is requested by CYBERDATA:

           VLOS, VLBCKG, VLA5S, BGCORE, LFTOVR, TOTAVL

VLOS is used by the Restart program to determine the size of Area 5.

    When VLOS = NONE, Area 5 is set to 0: No area is specially
    reserved for level 5 programs.

    When VLOS = STD, Area 5 is set to length VLA5S; an installation
    parameter.

    When VLOS = BCKG, Area 5 is set to the total length of allocat-
    able memory minus BGCORE (an area to be left available for back-
    ground programs) minus Areas 1-4 specified at install time.

For both the STD and BCKG options TOTAVL (total available) is set to
the full length of allocatable memory minus LFTOVR. LFTOVR is an
installation parameter (currently 280_{16}) that defines an area that will
be available for loading either DEBUG or SYSCOP when CYBERDATA has
taken the maximum memory.

The value of VLOS may be changed using the program VLOSOP. VLOSOP
is called with the following sequence:

| MI | VLOS | CR |

    System response: VLOS IN

    Enter one of the following:
        STD, BCKG, or NONE

    (Any other input results in the message DATA ERROR)
    The program then exits with the response VLOS OUT.

    The new value of VLOS takes effect at the next Autoload and Restart.

VLBG is used by the CYBERDATA start-up program to determine the number of
words to be requested from allocatable memory for the CYBERDATA buffer.

    When VLBG = NONE, TOTAVL words are requested.

    When VLBG = MUST, Area 5 is requested.

    When VLBG = RQST, an optimum number of words is requested. This
    figure is based on the NO. OF STATIONS and AVERAGE RECORD LENGTH
    parameters entered at start-up time.

The value of VLBG may be changed by using the program VLBGOP. VLBGOP is
called with the following sequence:

| MI | VLBG | CR |

    System response: VLBG IN

    Enter one of the following:

        xxxx, NONE

        xxxx, MUST

        xxxx, RQST

Where xxxx is the hexadecimal value desired for the BGCORE parameter.
(Any other input results in the message DATA ERROR) The program responds
with VLBG OUT. The new value of VLBG takes effect at the next Autoload
and Restart.

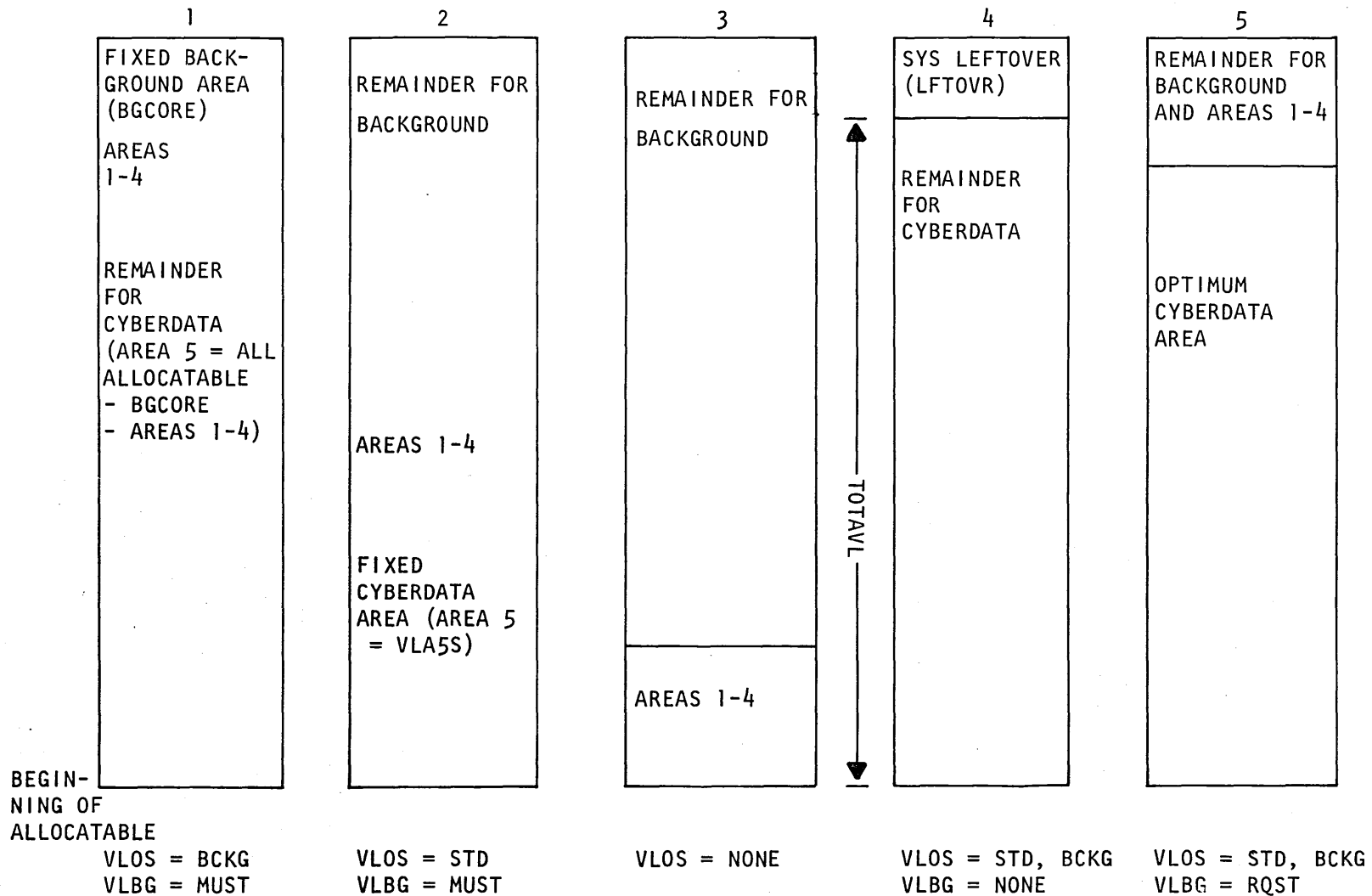# CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___VLOSOP___

PROGRAM NAME___ Select Startup Conditions___

The five examples on the following page show how allocatable memory would be partitioned, given the different values of the parameters VLOS VLOS and VLBG.

The program VLST can be called to give a printout of what the result of the different allocation schemes will be.

An example of allocatable memory as it can be partitioned is illustrated on the next page.

AA5641

1	2	3	4	5
FIXED BACK-GROUND AREA (BGCORE) AREAS 1-4 REMAINDER FOR CYBERDATA (AREA 5 = ALL ALLOCATABLE - BGCORE - AREAS 1-4)	REMAINDER FOR BACKGROUND AREAS 1-4 FIXED CYBERDATA AREA (AREA 5 = VLA5S)	REMAINDER FOR BACKGROUND AREAS 1-4	SYS LEFTOVER (LFTOVR) REMAINDER FOR CYBERDATA	REMAINDER FOR BACKGROUND AND AREAS 1-4 OPTIMUM CYBERDATA AREA

BEGINNING OF ALLOCATABLE

VLOS = BCKG VLBG = MUST	VLOS = STD VLBG = MUST	VLOS = NONE	VLOS = STD, BCKG VLBG = NONE	VLOS = STD, BCKG VLBG = RQST

TOTAVL

EXAMPLES OF PARTITIONING ALLOCATABLE MEMORY

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___VLSTRT - Cyberdata Start Up Program_____Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   The main functions of VLSTRT are:

   a. To request the number of active terminals and average record length from the supervisor stations.

   b. To calculate and allocate the required memory space for Cyberdata tables and disk-resident programs.

   c. To initiate the Cyberdata tables and buffers.

   d. To check if the selected disks are ready.

   e. To clear the terminal controllers, and initiate the clock interrupt line.

   f. To set the Cyberdata active flag (MIBVL).

   g. To call CLR to restart or freeze active batches, after power fail.

2. DETAILED DESCRIPTION

   VLSTRT is loaded by MSOS to AREA5 of the allocatable memory, at priority 6.

   The program performs the following steps:

   a. Saves the start address of VLSTRT for the RELEASE request, in order to release its own memory space.

   b. Reads the Terminal Type Table, via th GETTBL routine, into an internal buffer called TTB.

   c. Reads the first 8 words of the PRD which contain the distances to the start of the PRD sections, calculates PRD length and stores it in the table of the table lengths.

   d. Types: "NO. OF TERMINALS=", reads the reply from the TTY, and checks if it is valid, and if not, returns to the beginning of this step.

   e. Releases VLSTRT memory space.

   f. Calculates Cyberdata disk area on the system disk, according to the installation parameters: SIBSEC, SIESEC.

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___VLSTRT - Cyberdata Start Up Program___ Level 6

g. Generates table lengths table, i.e. puts the lengths of Cyberdata memory resident tables and buffers into the table lengths table.

h. Claculates the minimum memory space requirement (MINCOR).

$$MINCOR = 2000 + \sum_{i=1,13} 1;$$

1_1 = FBA length (no. of terminals + 1)
1_2 = TIQ's length
1_3 = ATT's length
1_4 = ALD's and TOPT's length
1_5 = ABT buffer length
1_6 = CWT length
1_7 = PRD length
1_8 = 0 (spare)
1_9, 1_{10}, 1_{11}, 1_{12} = TRT's lengths
$1_{13} = 64$

i. Gets average record length from the supervisor.

j. Calculates optimal memory space requirement (OPTCOR).

$$OPTCOR = MINCOR + N*R + 2000 + \sum_{i=1,k} C;*16 + C_{k+1}* \quad (N mod\ 16)$$

N = Number of terminals
R = Average record length
K = Integer value of N/16, i.e. N=16*K + (Nmod 16)
C_1= 200, C_2= 100, C_3= 70, C_4= 50

k. Calculates actual memory for SPACE request (SPACRQ)

$$SPACRQ = \begin{cases} TOTAVL & \text{if B.G (BACKGROUND) not required} \\ VLAS & \text{if B.G required} \\ min\ (OPTCOR, TOTAVL) & \text{if B.G is optional} \end{cases}$$

TOTAVL =   Total memory space available for Cyberdata (calculated by SPACE program at autoload time).

VLAS   -   AREA 5 size (decided by SPACE at autoload time, according to the memory space required for B.G)

AA5641

l.   Requests SPACRQ words via a SPACE request.

m.   If $SPACRQ \leq MINCOR + N \ast R + \frac{1}{2} (\sum_{i=1,k} C_i \ast 16 + C_{k+1} (N \bmod 16))$
     then prints the warning message:

     'MEMORY AVAILABLE UNDER EFFICIENCY LIMIT'

n.   If allocated memory begins before VLSTRT, then moves VLSTRT to
     the beginning of the allocated memory.

o.   Clears allocated memory from the end of VLSTRT.

p.   Sets table and buffer addresses in SST and EST.

q.   Builds ATT's and TOPT's.

r.   Reads the following tables from the system disk: ABT, CDV
     weights, PRD, TRT's.

s.   Builds ALD's.

t.   Sets disk state (last used and full).

u.   Deselects selected disks with write protect switch 'ON' and
     those which are not ready. Terminates VLSTRT is disk 1 is protected
     or not ready.

v.   Calculates Julian date according to the system date (AYERTO, AMONTO,
     ADAYTO) and puts it in EST.

w.   Reads the Statistics File, counts number of records and puts it in EST.

x.   Clears terminal controllers and initiates clock interrupt.

y.   Initiates Cyberdata clock in SST.

z.   Sets CYBERDATA active flag (MIBVL) and calls CLR program.

INTERFACE

1. ENTRY METHOD

    VLSTRT is invoked by the supervisor in one of the following ways:

    a.   Autoload the system.

    b.   Type (MI) VL (CR), when Cyberdata is not active.

    In either case, VLSTRT is invoked only when the 'VLOS' parameter is
    STD or BCKG.

    Control is transferred to VLSTRT from the dispatcher as a result of a
    schedule request from SPACE or MIPRO (MSOS program).

2. INPUT PARAMETERS

   a. No. of terminals - * or ASCII digits, received from the TTY, in the range 1 - 64.

   b. Average record length - up to 4 digits, received from the TTY, in the range 1-MAXREC (MAXREC is word 38 of EST and contains the maximum record length).

   c. The following entries of the SST: IBL, ATTLNG.

   d. The following entries of the EST: TRT lengths (words 1,2), MAXOCD, SFDA, ESSF, LTDDA, LDDDA, PRDDA, TRT disk addresses (words 17-20), CDVA, SPIL, disk logical units, WES's, MAXREC, TTBDA.

   e. The following externals (see 3 of ASSEMBLY CHARACTERISTICS): STTLNG, SIABTW, INTLIN, VLBCKG, VLA5, TOTAVL, VLILIN, VL2SCN, HORMIN, SIBSEC, SIESEC.

3. EXIT METHOD

   a. Normal exit is to the CLR program (by ENTPRG).

   b. Error exit is to the dispatcher (by: JMP- ($EA)).

4. OUTPUT PARAMETERS

   a. The following messages (see Cyberdata System Reference chapter 6):

      1) No. of STATIONS=

      2) INSUFFICIENT MEMORY FOR REQUESTED CONFIGURATION, nnnn WORDS ARE MISSING.

      3) ** WARNING ** MEMORY AVAILABLE UNDER EFFICIENCY LIMIT

      4) TERMINAL CONTROLLER nn REJECTED

      5) NO CLOCK INTERRUPT

      6) CYBERDATA TERMINATED

      7) DISK I/O ERROR

      8) AVERAGE RECORD LENGTH=

      9) 'ES' NOT FOUND IN STATISTICS FILE

      10) 970-32 KEYSTATIONS NOT AVAILABLE ON THIS SYSTEM

      11) DISK n IS DESELECTED

      12) DISK 1 IS DESELECTED

AA5641

    b. The following SST entries: SLOCK, PRDA, THCRT, THSLFS, FRPLL, FRPUL, FRPP, ATTB, CATT, TIQ, CLOCKM, CLOCKS.

    c. The following EST entires: ACWT, MAXCDV, TRT's memory address, FBA, ASRCHB, ABTADR, DSKST, TDATE.

    d. The following externals (see 3 of ASSEMBLY CHARACTERISTICS): FREEBK, MIBVL.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. SST,EST and ATT EQUATES are explained in the Cyberdata Software User's Guide.

    b. Low core EQUATES are explained in MSOS 4 Reference Manual.

    c. MAXTN - Maximum number of terminals.

    d. TOPT1L - Selfscan TOPT length.

    e. TOPT2L - CRT TOPT length.

    f. BLOKL - ALD's block length.

    g. BLOKF - Log 2 of BLOKL (for masking purposes).

    h. C1, C2, C3, C4 - Number of free memory words per each terminal in the internals: 1-16, 17-32, 33-48, 49-63 correspondingly.

    i. D1 - Start number of free memory words.

    j. D2 - Number of free memory words for terminals 1-16.

    k. D3 - Number of free memory words for terminals 1-32.

    l. D4 - Number of free memory words for terminals 1-48.

2. MACROS AND SUBROUTINES

    a. The following system MACROS are used: FREAD, FWRITE, RELEASE, SPACE.

    b. The following internal subroutines are used:

        1) GETTBL - subroutine to read the memory resident tables from the system disk to the allocated memory.

        2) READIT - subroutine to read up to 4 digits from the TTY.

        3) PUTMSG - subroutine to print messages.

        4) DISKIO - subroutine to read or wrote a test word from a disk.

3. PROGRAM DEPENDENCIES

   a. ENTRY POINTS

      None.

   b. EXTERNAL LIST

      STTLNG - bits 0-7 = maximum special tests.
               bits 8-15 = maximum CDV tests.
      SIABTW - ABT buffer length (words).
      INTLIN - screen start position of the INTERROGATE line.
      FREEBK - pointer to the ALD free chain.
      MIBVL - Cyberdata active flag.
      VLBCKG - background switch.

$$VLBCKG = \begin{cases} 1 - \text{BG required} \\ 0 - \text{BG optional} \\ 1- \text{BG not required} \end{cases}$$

      VLA5   - length of system allocatable area 5.
      TOTAVL - maximum number of words available for Cyberdata.
      VLILIN - Cyberdata interrupt line number.
      VL2SCN - Selfscan driver entry point.
      AYERTO, AMONTO, ADAYTO - year, month and day, entered during
                                    start-up procedure.
      SIBSEC - Cyberdata first sector on the system disk.
      SIESEC - Cyberdata last sector on the systemdisk.

   c. ROUTINES CALLED VIA TRANSFER VECTOR

      GETBIN, DISPLY, CLCDWA, ENTPRG

   d. DATA AREAS AND SYSTEM TABLES USED

      SST, EST, FBA, TIQ, ATT's, ALD's ABT, CWT, PRD, TRT's.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE

   1290_{10} words.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Not reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___VLSTTS - Start Up Status Report___   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    VLSTTS is an overlay called in by MIPRO.
    CALLING:

           MI
           VLST CR
    The resulting printout will be:

           AREA LEFT FOR SYSTEM USE XXX1
           REQUESTED BACKGROUND XXX2
           CYBERDATA ACTIVATION XXX3
           BACKGROUND ACTIVATION XXX4

           VLOS BCKG CYBD BCKG
           ACTV ACTV CORE CORE

           STD NONE xxx5 0000
               MUST xxx6 xxx7
               RQST DYNM UNDF

           NONE ---- 0000 xxx8

           BCKG NONE xxx5 0000
                MUST xxx9 xxx2
                RQST DYNM UNDF

           WHERE   xxx1 = LFTOVR
                   xxx2 = BGCORE
                   xxx3 = Value of VLOS (STD, NONE, or BCKG)
                   xxx4 = Value of VLBG (NONE, MUST, or RQST)
                   xxx5 = TOTAVL (All of allocatable -LFTOVR)
                   xxx6 = VLA5S (CYBERDATA standard AREA S length)
                   xxx7 = All of allocatable - VLA5S - Areas 1 to 4
                   xxx8 = All of allocatable - Areas 1 to 4
                   xxx9 = All of allocatable - BGCORE - Areas 1 to 4

2. DETAILED DESCRIPTION

    VLSTTS save its address for release at completion. Each core value needed
    in the report is loaded in A. The internal subroutine HEXASC converts the
    value to ASCII with results in A and Q. Then the internal subroutine STORE
    is used to put the values from A and Q into the print buffer. An FWRITE
    macro is used to output the buffer to the standard comment device. When the
    output is complete, VLSTTS releases its own memory and jumps to the dispatcher.

INTERFACE

1. ENTRY METHOD

    Execution begins at first instruction of this system loaded overlay.

2. INPUT PARAMETERS OR CONDITIONS

    No parameters are passed to the program.
    The operator enters MI
                              VLST

3. EXIT METHOD

    Release core and jump to dispatcher.

4. OUTPUT PARAMETERS OR CONDITIONS

    Status report is output to system comment device.

ASSEMBLY CHARACTERISTICS

1. EQUATES

    LOCORE

    LPMASK (2)

2. MACROS USED

    FWRITE
    RELEAS

3. PROGRAM DEPENDENCIES

    a. SYSDAT   LFTOVR - Amount of core left for Debug or Syscop
                  BGCORE - Required amount of core for necessary background
                  VLOS   - Cyberdata active flag
                  VLBCKG - Cyberdata background flag
                  TOTAVL - Total available core for Cyberdata
                  VLA5S - Cyberdata standard area 5 length
                  N1   N9
                  N2   N10
                  N3   N11
                  N4   Length of allocatable   N12
                  N5   core for areas 1-15   N13
                  N6   N14
                  N7   N15
                  N8

                  AREAC - Beginning address of allocatable .

AA5641   PRINTED IN THE USA

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME__ VLSTTS - Start Up Status Report__ Level 6

4. DATA AREAS USED

Internal print buffer.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$146_{16} = 326_{10}$ words.

2. RELOCATION TYPE

Run anywhere.

3. ENTRANCY CLASSIFICATION

Not reentrant.

DOCUMENT CLASS ___VLTYPE___

PROGRAM NAME___Print and Change Terminal Type Table___

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

VLTYPE is an overlay called in by the manual interrupt processor MIPRO.
It accepts the following commands from the system operator console:

*D	Print directory of terminal types
*P,N1,N2	Print type table (terminals N1 through N2)
*P	Print type table (terminals 1-64)
*C,N1,T1	Change type of terminal N1 to type T1
*Z	End of VLTYPE

2. DETAILED DESCRIPTION

VLTYPE saves the address where it was loaded so that it can release core
at completion. Using an FWRITE macro it outputs "VLTP IN" to the system
comment device. Using a FREAD it puts a 4 word input against the system
command input device and waits until the operator inputs one of the commands
listed above. If an operator entry error was made, VLTYPE outputs an "ILLEGAL
REQUEST" or "XX ILLEGAL TYPE" message.

First VLTYPE checks for *Z input. If so, it outputs "VLTP OUT" to the
system comment device and releases its core.

A print directory *D request results in an output to the system comment
device of all the possible terminal type definitions. (See attachment A.)

A *P,N1,N2 request causes a printout of the terminal type for terminals
N1 through N2. In order to do this, first the parameters N1 and N2 are
checked for validity. Then the terminal type table, which is stored on
disk, is read into a buffer internal to VLTYPE and printed on the system
comment device one line at a time. A *P command is handled the same way
except that the N1 is set by VLTYPE to 0 and N2 to max. legal value for
the system depending on the no. of controllers.

A *C,N1,T1 request causes a terminal type to be changed. If the input
parameters are valid and the appropriate display driver or keyboard
conversion routine has been loaded in the system, VLTYPE reads the
terminal type table into its buffer, changes it, writes the table back
to disk, and prints the new type on the system comment device.

At the conclusion of each command except *Z VLTYPE outputs "VLTP IN" and
waits for the next command from the operator.

INTERFACE

1. ENTRY METHOD

Execution begins at the first instruction of the system loaded overlay.

2. INPUT PARAMETERS OR CONDITIONS

Operator hits M1 and types VLTP.

DOCUMENT CLASS ___VLTYPE_____

PROGRAM NAME___Print and Change Terminal Type Table_____

No parameters are passed to the overlay.

The address where it is loaded is stored immediately preceding its first instruction.

3. EXIT METHOD

Release core and go to dispatcher.

4. OUTPUT PARAMETERS OR CONDITIONS

SYSTEM TYPE TABLE MODIFIED and/or REQUESTED REPORT PRINTED

ASSEMBLY CHARACTERISTICS

1. EQUATES

a. LOCORE

LPMASK (2)
NZERO ($12)
TEN ($46)
SST ($47)
DISPCH ($EA)

b. CONSTANT

L (4)   Input buffer size in words

c. EST

MTTYPE (41)

2. MACROS USED

TYPE   (See Attachment B)
FWRITE
FREAD
RELEAS

3. PROGRAM DEPENDENCIES

NMONI
DCORE
RW

4. DATA AREAS USED

All data areas used are internal to the program.

38

DOCUMENT CLASS ____VLTYPE____

PROGRAM NAME____Print and Change Terminal Type Table____

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   $1BC_{16} = 444_{10}$ words

2. RELOCATION TYPE

   Run-anywhere

3. ENTRANCY CLASSIFICATION

   Not reentrant

ATTACH ANALYTICAL SUPPLEMENTS

Attachment A: Terminal Type Definitions
Attachment B: TYPE Macro skeleton and MTTYPE table setting

DOCUMENT CLASS VLTYPE

PROGRAM NAME Print and Change Terminal Type Table

Attachment A

Terminal Type Definitions Directory

TYPE	K/B	DSP	MODE
00	029	SCN	SUP
01	029	SCN	STD
02	029	CRT	SUP
03	029	CRT	STD
04	TTY	SCN	SUP
05	TTY	SCN	STD
06	TTY	CRT	SUP
07	TTY	CRT	STD
08	ADD	SCN	SUP
09	ADD	SCN	STD
10	ADD	CRT	SUP
11	ADD	CRT	STD
12	TTN	SCN	SUP
13	TTN	SCN	STD
14	TTN	CRT	SUP
15	TTN	CRT	STD

```
Where:   K/B = 029   Keypunch (029)
               TTY   Teletype
               ADD   Keypunch (029) with numeric cluster
               TTN   Teletype with numeric cluster

         DSP = SCN   970-32  display
               CRT   970-480 display

        MODE = SUP   Supervisor
               STD   Standard
```

40

```
                        VFD   N5/0,N1/0,N2/3,N1/0,N1/0,X6/'TN'
                        EIF
                        IFA   'TP',EQ,14
                        VFD   N5/0,N1/1,N2/3,N1/1,N1/0,X6/'TN'
                        EIF
                        IFA   'TP',EQ,15
                        VFD   N5/0,N1/1,N2/3,N1/0,N1/0,X6/'TN'
                        EIF
                        EMC
```

Calling sequence:

        TYPE   TN,TP

   Where   TN = Terminal number
              TP = Type code as listed in attachment A

MAX legal value for the system depending on the no. of controllers.

4.   SYSTEM TYPE TABLE MODIFIED and/or REQUESTED REPORT PRINTED

     TYPE Macro skeleton and MTTYPE table setting

42   Standard setting:

            MTTYPE is set up initially as allows

                Terminal 0 - type 00 (keyboard/printer supervisor)
                All other terminal - type 01 - 029 SCN STD

42

DOCUMENT CLASS ___VLTYPE___

PROGRAM NAME___Print and Change Terminal Type Table___

Attachment B

TYPE Macro Skeleton

```
TYPE   MAC  TN,TP
       IFA  'TP',EQ,0
       VFD  N5/0,N1/0,N2/0,N1/1,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,1
       VFD  N5/0,N1/0,N2/0,N1/0,N1/0,X6/'TN'
       EIF
       IFA  'TP', EQ,2
       VFD  N5/0,N1/1,N2/0,N1/1,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,3
       VFD  N5/0,N1/1,N2/0,N1/0,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,4
       VFD  N5/0,N1/0,N2/1,N1/1,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,5
       VFD  N5/0,N1/0,N2/0,N1/0,N1/0,X6/'TN'
       EIF
       IFA  'TP', EQ,6
       VFD  N5/0,N1/1,N2/1,N1/1,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,7
       VFD  N5/0,N1/1,N2/1,N1/0,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,8
       VFD  N5/0,N1/0,N2/2,N1/1,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,9
       VFD  N5/0,N1/0,N2/2,N1/0,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,10
       VFD  N5/0,N1/1,N2/2,N1/1,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,11
       VFD  N5/0,N1/1,N2/2,N1/0,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,12
       VFD  N5/0,N1/0,N2/3,N1/1,N1/0,X6/'TN'
       EIF
       IFA  'TP',EQ,13
```

41

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME_____ VL2CRT - 970-480 Display Driver _____ Level 06

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   VL2CRT is the CRT display driver. There are three program entry points:

   a. CDATA   - initiator

   b. VL2CRT   - continuator

   c. CONCSR   - convert binary CRT address to two 8-bit display codes.

   There are also 15 entry points which are equates to two 8-bit display codes
   contain values describing the screen layout, the top of the free allocatable/
   display blocks string, and the maximum number of blocks allocated per terminal.
   All CRT display functions build a call to this driver. The initiator part of
   the driver transfers the call parameters to the TOPT (Terminal Output Table)
   for reentrancy purposes. The data to be displayed and screen addresses are
   moved to an ALD (allocatable display buffer). The driver returns to the
   caller when all the characters for display have been moved to the ALD. The
   continuator (VL2CRT) which actually performs the output is called by VLINT
   for each CRT terminal when a clock interrupt occurs.

2. DETAILED DESCRIPTION

   INITIATOR

   (CDATA). Flags in TOPT/14 are initialized. The 5-word parameter/data block
   is moved to TOPT/8-12. The return address is stored in the temporary storage
   area of the SST.

   (CD21). <u>For a Parameter Block.</u> The return address is adjusted for the
   actual number of parameters passed. If I/O bit processing is required, the
   user's I/O bit is checked. If off, it is set and a flag is also set in
   TOPT to indicate that the I/O bit must be cleared on exit. If a start of
   display screen address is supplied by the caller, it is converted to output
   format and stored in ATT/RQ. If it is not supplied, it is taken from TOPT
   and the screen address used is updated to reflect the position at end of
   display. Go to CD4.

   (CD3). <u>For a Data Block.</u> The block previously moved to TOPT is scanned
   for the first nonzero word. When such a word is encountered it is stored
   in ATT/RQ and set to zero in TOPT. Otherwise, exit via EXDSR.

(CD4). If first character flag is not on, go to CD8 to advance to the next word in the string. Otherwise,

(CD41). If a display string already exists for this terminal, go to CD5 to find the end of string. Otherwise, when there is no block (ALD) assigned to the terminal, and the blocks are occupied by other terminals, stay in WAIT until a block is released the released block is linked to the group of blocks assigned to the requesting terminal, and if required link the terminal TOPT to the thread of CRT TOPT's. Then fix pointers for storing data in the block (STR PTR) and extracting characters for display (TOPSTR). When there is a block assigned to the terminal, check if there is an empty word. When an empty word is found, its address is stored in STRPTR, and go to CD6; otherwise, go to CD42 to look for another block.

(CD6). Move Current Word at ATT/RQ to ALD Block. The word in ATT/RQ is moved to the location whose address is in STRPTR.

If the information following the (CDATA) call was data, then the first character flag is cleared. Go to CD3 to get the next word.

If the information was parameters and the first character flag is not set, go to CD7. If it is set, move the repeat bit and number of parameters from TOPT(8) to TOPT(14). Expand the character count in TOPT(8) to a full word for ease in using as counter.

If the caller's data starts on the least significant byte, increment this character count by 1 and set MS byte to filler when moving to ATT/RQ.

(CD63). Move unmodified word to ATT/RQ. Clear possible error flag on MS bit of byte.

If the repeat bit is not on, advance the data address in TOPT(9) and go to CD8 to advance to the next word in the string.

(CD7). Decrement by 2 the number of characters to be output. (TOPT/PARA5). If still positive go to CD63 to get next word. If the result is zero, go to CD73 to check if output of final cursor is required. If result is - 1, modify the last character moved to ALD to be filler. Go to CD73.

(CDEXIT). Clear the I/O bit on caller's buffer if required, and return to caller via EXDSR.

(CD73). Check for Final Cursor. If a final cursor not required, go to CDEXIT. Otherwise get it from parameter list. Convert to output format and store in ATT/RQ.

Update current screen position in data or interrogate area with final
cursor value.

(CD8). Move To Next Word In String. Increment string pointer STRPTR.
If not end of block, go to CD6 to move current word to location indicated
by STRPTR.

If end of block, go to CD42 to get another block.

CONSCR

This subroutine converts a binary screen address to 2 bytes. Both input and
output are in A register.

        Byte Zero: B7   = 0
                    B6-5 = 11
                    B4-0 = LINE NUMBER
        Byte One: B7   = 0
                    B6   = 1
                    B5-0 = character position within line.

PROGRAM DESCRIPTION

CONTINUATOR

(VL2CRT). PASS2 switch is set to FINAL/NON FINAL pass according to present
value in OPSW1X is set to zero/nonzero for output of one/two characters
per interrupt.

(VL2CR1). The address of the top of the CRT thread in SST (SST/TOPCRT) is
set as the address of the previous CRT PRVCRT.

Get top of the thread.

(VL2C1). Check the thread for FFFF (end of thread). If not FFFF, go to
VL2C11 to continue. Check PASS2 switch. If zero, return to caller via
entry point. If not zero, set to zero and loop back to VL2CR1 to make
another pass through the thread.

(VL2C11). Clear LBFLAG (Loop-Back Flag). Check if terminal requires display of system-lock. If not, check if message is waiting for display and if it has been initiated. If not, go to VL2C1X to display data; otherwise, go to MSGOPS to display the message. If system locked, check if lock output already initiated. If not, go to VL2C1X. If yes, use lock flags as index to lock-table. Extract lock character and output. Decrement lock flags and check for zero. If zero, go to RELBK2 to see whether terminal may now be unthreaded. Otherwise, go to VL2C13 to advance to the next terminal in the thread.

(VL2C1X). Select byte for output at the top of string.

(BYTE0). If BYTE-ZERO is filler, go to BYTE 1. Otherwise, set Q to WES and output. If BYTE-1 is filler, go to BYTE 11 to advance to next word.

If BYTE-1 is not filler, go to BYTE01.

(BYTE01). Flip BYTE indicator in TOPT/BLOKS to point to BYTE 1.

(VL2C13). Move address of current terminal's TOPT to PRVCRT - previous CRT. Pick up thread to next terminal. Go to VL2C1 to check for end of thread.

(BYTE1). Get BYTE-1 and check for filler. If filler, set LBFLAG and go to BYTE11 to advance to next word. If not filler, set Q to WES and output.

(BYTE11). Increment top of string in TOPT. Set byte indicator to byte zero. If top of string now points to end of block, go to RELBLK to release the block. If top of string does not point to a word whose contents are zero, store its address as top of string and go to RELBK2.

(RELBLK). Release Block. Release block by storing its address as NEXT FREE BLOCK in FREEBK and by moving the previous contents of FREEBK to it. Decrement the number of blocks allocated to this terminal by 1. Store link word of block released in top of string.

(RELBK3). If top of string zero, go to RELBK3 to see if terminal may be unthreaded. Otherwise, check LBFLAG. If on, loop back to VL2C1X to output next byte. If not on, go to VL2C13 to advance to next terminal.

(RELBK3). Check if lock output is required. If not, go to RELBK5 to check if message output is required. Otherwise, set lock flags to indicate lock output in progress and go to VL2C13 to advance to the next terminal.

(RELBK5). If the message flag (FLDSI,I) is set, set flag to indicate that message has been initiated (COL,I) and go to VL2C13 to advance to next terminal, else go to RELBK4 to unthread the terminal.

(RELBK4). <u>Unthread Terminal</u>. Terminal is unthreaded by storing contents of its thread word (ATT/TOPT) as thread word of previous terminal whose address is in PRVCRT. Own thread is set to zero. Go to VL2C1 to check if the thread to the next terminal is FFFF.

(REJ). <u>Reject Routine.</u> A reject counter is updated. Control passes to VL2C13 to advance to the next terminal.

(MSGOPS). Check ATT/7,6,5 when a positive word is found extract the corresponding code from MSGCODE table and output it. Set bit 15 of that word and go to VL2C13 to advance to next terminal. When there is no positive word in ATT/7,6,5 check the first word of the message to find whether the first-display-flag is set. If it is not set, display a blank set the flag, enter two to character count and go to VL22C13. When the first display is set, extract and display the next character. If it was the most significant character mark it by setting bit 14 restore the word and go to VL2C13. If it was the least significant character check the message word count as follows:

a. Word count is 2   - enter 5.

b. Word count is less than 7 - increase it by 1.

c. Word count is 7   - enter 15.

   Shift word count to the most significant byte, set bits 14, 15 and restore to ATT/2.

d. When word count is 15, clear the message flag (FLDST,I) and go to RELBK2 to check for end of string.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME__ VL2CRT - 970-480 Display Driver__   Level 06

INTERFACE

CDATA:

    I = Current ATT
    The call parameters may be of 2 types.
    Type 1 where modifier bit M is zero.

M

0	DATA CHAR 0	DATA CHAR 1	0
	2	3	1
	4	5	2
	6	7	3
	8	9	4

Here the parameter block is used only for the transfer of data. A full 5-word block is assumed to follow the call. Data is terminated by the first negative word in block. Any all-zero word is disregarded.

Type 2 where modifier bit M is one.

15	13	12	11	8 0	
M	OPT	R	B	NO. OF CHARS	0
ADDRESS OF DATA					1
I/O BIT ADDRESS					2
SCREEN ADDRESS FOR O/P					3
FINAL CURSOR VALUE					4

The first two parameter words must be supplied. The last three are optional. The driver returns to the location following the last parameter supplied.

        WORD ZERO   BIT 15   = MODIFIER BIT
                       14-13 = NUMBER OF OPTIONAL PARAMETERS INCLUDED
                       12   = REPEAT BIT
                       11   = BYTE ADDRESS 0/1 ON WHICH DATA STARTS
                       8-0 = NUMBER OF CHARACTERS FOR DISPLAY

WORD ONE   CORE ADDRESS OF DATA FOR DISPLAY
WORD TWO   ADDRESS OF WORD CONTAINING I/O BIT
WORD THREE   SCREEN ADDRESS FOR DISPLAY
WORD FOUR   CURSOR ADDRESS AT END OF DISPLAY

MODIFIER BIT 0/1

0   Selects data block
1   Selects parameter block

OPT 0/3

Number of optional parameters from zero to three

REPEAT BIT 0/1

1   Repeatedly outputs same 1 or 2 characters from address defined in
     parameter WORD 1 'N' times where N is defined by "NO OF CHARACTERS"
     in parameter word zero.

0   Outputs data from consecutive core locations ad defined by
     parameter words 0/1.

BYTE 0/1

Defines byte at which data at address in parameter WORD 1 starts

0   Byte at B15
1   Byte at B7

NO OF CHARS

Number of characters to be output

ADDRESS OF DATA

Core word address of data.

Data is assumed to be 7-bit ASCII packed 2 characters per word.

I/O BIT ADDRESS - OPTIONAL PARAMETER

If zero disregarded. If not zero, this is the core address of the I/O bit word
in user's buffer. The driver sets the I/O bit for the duration of its operation
and restores its original value on completion.

SCREEN ADDRESS FOR O/P - OPTIONAL PARAMETER

Address at which data is to be displayed. If this parameter is not given the
screen address is taken from the TOPT. Two screen addresses are maintained
in the TOPT, one for the DATA portion of the screen and one for the INTERROGATE
area. The appropriate address is selected according to the data definition
setting in the TOPT.

Where this parameter is not given, the screen address in TOPT is updated to reflect the new address at completion of display. Where it is given, the address in TOPT is not updated.

NOTE: The driver assumes the screen address maintained is the address to which the last character was output. It increments this address by 1 <u>before</u> outputting. For this reason, if screen addresses are given in parameters 3 or 4 they should be the true address minus 1.

<u>FINAL CURSOR VALUE</u> - OPTIONAL PARAMETER

Screen address at which cursor is to be left at conclusion of output. If not given, the cursor will be left on the last character output.

Where this parameter is given, it replaces the current screen address in TOPT, in Data or Interrogate area.

VL2CRT: Entry is by return jump. No input parameters.
CONCSR: Entry is by return jump.
          A = binary value of jump cursor position.

3.   EXIT METHOD

CDATA: Exit is via the display programs exit subroutine EXD6R.
          All characters for display have been moved to the ALD, but the characters have not been output when control is returned to the caller.

CONSCR: Exit is jump back to caller. A = 2 cursor screen address in output format.

VL2CRT: Exit is jump back to caller.
          1 or 2 characters are output to screen each interrupt time.

DOCUMENT CLASS __CYBERDATA__

PROGRAM NAME __VL2CRT - 970-480 Display Driver__   Level 06

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ZERO ($22)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        EXDSR ($8F)
        WAIT ($B0)
        DISP ($EA)

    b. SST

        TOPCRT (4)
        ATT (11)
        TMP (20)
        STKPTR (SST + TMP)
        STRPTR (SST + TMP + 1)
        RTRYFG (SST + TMP + 2)
        TEMP (SST + TMP + 3)

    c. ATT

        MODE (2)
        TOPT (33)
        IOADDR (64)
        RQ (65)
        COL (2)
        JOB (5)
        FLDST (15)

    d. TOPT

        THREAD (0)
        EWES (1)
        CURSRD (3)
        CURSRI (4)
        PARAS (8)
        CAR1F (11)
        TOPSTR (13)
        BLOKS (14)

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME ___VL2CRT - 970-480 Display Driver___   Level 06

e. CONSTANTS

C (1)
BLOKF (3)
BLOKL (8)
MODIF (10)
FLDNAM (4)
LINES (10)
LINCAR (48)
DATLOW (LINCAR-1)
DATUP (Lines * LINCAR-LINCAR-1)
INTADR (DATUP)

2. MACROS USED

None.

3. PROGRAM DEPENDENCIES

ENEXSR
NDISP
CMISC1

4. DATA AREAS USED

SST
ATT
TOPT
ALD

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$1F6_{16} = 502_{10}$ words.

2. RELOCATION TYPE
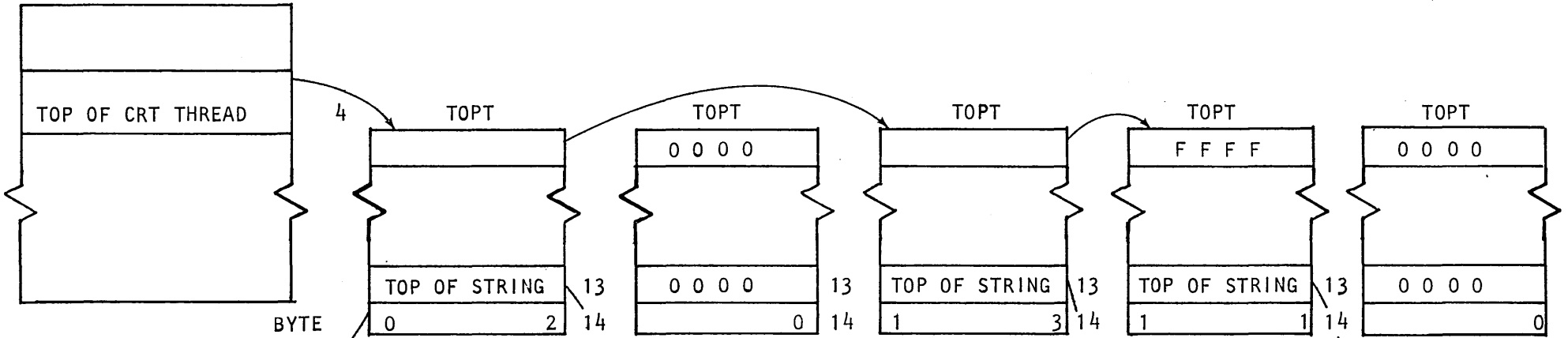
Loader relocatable.

3. ENTRANCY CLASSIFICATION

CDATA: Dormant reentrant
VL2CRT: Reentrant
CONSCR: Not reentrant

ATTACH ANALYTICAL SUPPLEMENTS

Attachment A: ALD Control Diagram

ALD CONTROL DIAGRAM

SST

TOP OF CRT THREAD   4   TOPT   TOPT   TOPT   TOPT   TOPT

                                                        0 0 0 0   F F F F   0 0 0 0

                                      TOP OF STRING 13   0 0 0 0 13   TOP OF STRING 13   TOP OF STRING 13   0 0 0 0

BYTE   0   2 14   0 14   1   3 14   1   1 14   0

NOTES:

1. TOPT (13, 14) define
   the top of string.

2. Characters are added
   to end of string and
   output from top of
   string.

3. First all zero word
   marks the end of
   string.

4. The No. of blocks in
   string is maintained
   in order to facilitate
   control of the amount
   of core allocated to
   a given terminal.

NO. OF BLOCKS
IN STRING

A	B
L I N K	

C	D
E	F
0 0	0 0
0 0 0 0	

K	FILLER
0 0	0 0
0 0	0 0
0 0	0 0

FILLER	B
C	D
L I N K	

E	F
G	H
I	J
L I N K	

FILLER	3
C	D
E	F
0 0 0 0	

ATTACHMENT A

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

VL2SCN is the 970-32 display driver. Each time the terminal controller
interrupts the system, VLINT checks the output thread for the 32-character
displays. If the thread is not empty, VLINT calls VL2SCN to display up to
two characters on each terminal queued for output. The first (leftmost)
character in a string requires that an address be sent before the character.
The following consecutive characters do not require an address. (If more
than 32 characters need to be sent, the new character is displayed in the
32nd position. This procedure is set by the way MCHAR stores overflow char-
acters in the output buffer.) The 'clear display' and 'bell' codes are
checked for before VL2SCN looks for data to be sent. After two passes have
been made through the output thread, control is returned to VLINT.

2. DETAILED DESCRIPTION

First VL2SCN initializes its counter/pointers and sets 1 to the top TOPT
on the thread. It checks TOPT/2/15 to see if a clear display request was
made. If so, the clear code is sent via the closed subroutine DSP300, the
clear bit is reset, and the last character output + 1 (TOPT/3/0-7) is set to
32 so that the next character output will be preceded by an address. Then
control goes to DSPZ5 where the next item on the thread is checked. When the
end of thread is reached, another pass is made through the thread so that two
characters made be output to each display for each controller interrupt. (If
an address needs to be sent, it is considered as one of the characters.) After
two passes, the PASCNT flag is cleared and control returns to VLINT.

If a clear display request is not made, a check is made for a bell request.
If so, the bell code is sent via DSP300, the 'bell' bit (TOPT/2/14) is reset,
and control goes to DSPZ5 where the next item on the thread is checked.

If no clear display or bell was requested, check if the screen position of
left most character for output (TOPT/3/13-8) is 32. If it is, the TOPT is
removed from the thread and VL2SCN goes on to the next terminal in the thread.

If the output for the terminal is not yet complete, check if screen position
of last character output + 1 is beyond the right-most position of the screen.
If it is, a flag (CARPOF) is set so that an address will be sent before the
next character to be output. The data definition bits (00 = data, 01 = inter-
rogate, 10 = idle/supervisor) are extracted from TOPT/2/7-8. The data defini-
tion is used as an index to the limit table to calculate the position of the
leftmost character for output.

DOCUMENT CLASS ___CYBERDATA___

PROGRAM NAME___VL2SCN - 970-32 Display Driver_____Level 06____

The position of the next character is used along with the overflow count to
calculate the next character position in the TOPT (stored in SOFTPT). The
most significant bit of the 8-bit character is checked. If set, the character
has not been output and control goes to DSP130, where the CARPOF flag is used
to see if an address must be sent. Either the address or the character is
sent via DSP300. When a character is sent, the not sent bit is cleared, the
last character output + 1 (TOPT/3/0-6) is updated and control goes to DSPZ5
where the next item on the thread is checked. If the most significant bit
of the character is not set, advance to the next position in the TOPT and
screen and increase leftmost position if required. Then if HDWPNT points
position 32, go to DSP405 to check the position of last character output + 1.
If not position 32, go to DSP74 ro DSP80 to fix the SOFTPT according to DATA
flag.

INTERFACE

1. ENTRY METHOD

   Return jump

2. INPUT PARAMETERS OR CONDITIONS

   None

3. EXIT METHOD

   Jump indirect through entry return to caller.

4. OUTPUT PARAMETERS OR CONDITIONS

   None

ASSEMBLY CHARACTERISTICS

1. EQUATES

   a. LOCORE
      LPMASK   (2)
      NZERO   ($12)
      ZERO   ($22)
      ONEBIT   ($23)
      ZROBIT   ($33)
      THSLFS   ($4C)

    b. TOPT

        CHARIN   (4)

    c. CONSTANTS

BELL	(4)	Bell output code
CLRALL	($19)	Clear display output code
ADCHPT	($60)	Address bits for output
DATAL0	(18)	Data length of data area
DATAL1	(29)	Data length of interrogate area
DATAL2	(32)	Data length of idle/supervisor area

2. MACROS USED

    None

3. PROGRAM DEPENDENCIES

    None

4. DATA AREAS' USED

    TOPT

    SST

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $A9_{16} = 169_{10}$ words.

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Not reentrant.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___ VMM - Verify Mode Processor _____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    VMM processes characters entered in Verify mode. The program exits to
    CLRSR, or EORVR if at end of field and there are no verifiable fields left
    in the current record. If a mismatch occurs, the entered character is
    saved. IDK mode is set and exit is via ERROR.

    There are ten additional entry points contained in VMM. They are:
    VMM1, RVFVER, VMM2, NOVAL, EORTST, ENDOF1, SETBEG, MATCH, GETNZ.

2. DETAILED DESCRIPTION

    (VMM). Save the current character in ATT/ATEMP. Check ATT/SW2/15 to see if
    the first nonfill character in the record has been found. If not, (GETNZ)
    check the Recording mode for variable. If variable, check for automatic
    function to look for right-justify space fill for alpha fields or left-
    justify zero fill for numeric fields. Set up fill character and set ATT/
    SW2/15 (first nonzero flag). Call subroutine GBYTE (GSE). Test the
    character against the fill character. If it is the fill character, set
    ATT/SW2/5 (SKIP needed at EOF), check for end of field, update character
    counts, and go get next character. If the end of field is reached with
    no nonfill character found, clear ATT/SW2/15 and go to MISS. If a nonfill
    character is found or when the Recording mode is not variable, (TEST),
    check the Entry mode character against the Verify mode character. If it
    does not match, (MISS) display sign needed error if field data type
    (ATT/FDTP/12 = 1) signed and sign was not entered; illegal character error
    if sign entered when not required; or, if not a sign error, store the
    verify character entered at ATT/MISCH, change the mode to IDK (ATT/CMODE/14 = 1)
    and display a mismatch error. In both mismatch cases, exit is via ERROR (GSE).

    (MATCH). Update character pointers, display the character and check for
    end of field. If not at end of field, go to CLRSR to clear the exits stack
    and lockout. If at end of field, make sure sign was entered if necessary
    and check for forced skip at end of field, either if ATT/SW2/5 on, or
    Boundary Check required in the format. If so, go to EMMSKP (EMM) to set
    skip key required and IDK mode. If no forced skip, go to ENDOF1.

    (ENDOF1). If revalidation is required (ATT/SW2/13 = 1), call VALDTE.

PAGE NO. ___771___

AA5641

(NOVAL) (RVFVER). If verification is required (ATT/SW2/14 = 1), clear
the reverify required flag, (SETBEG) set pointers to beginning of field,
and check current field to see what the verification type is (CHKVRY).
If not sight verify, call DISPLY to clear the field and display the
column number. (SETFLD) clear the first nonfill skip needed next flags,
regenerate the display if required (ATT/SWITCH ≠ 0), or set the display
for beginning of field, and display column no. (END) if ignore first
skip flag is on (ATT/SW1/9), set IDK mode. (END1). Exit via CLRSR.

(SIGHT). If sight verify is required, set IDK mode and set CONT key
must be next bit (ATT/SW1/5). Either regenerate the display or set
display for beginning of field, display the field, clear the regenerate
switch, and go to END. If no reverification is required, the Terminal mode
is Pseudo Verify, and this is an insertion record, exit to TSTREC (EMM)
to handle the inserted record via the COR key when this is the last
field in the record or to set up for the next field when this is not
the last field. If no verification is required, the terminal mode is
Pseudo Verify, and this is not an insertion record, set all pointers
to the beginning of the next field (SETBEG), set the display to beginning
of field and exit via CLRSR. If no reverification is required and the
terminal mode is not Pseudo Verify, display end of field and test for
end of record.

(EORTST). If this is the last field in the record, increment the record
count (ATT/RCD), set the regenerate display flag and jump to routine EORVR.
If this is not the last field in the record, return jump to BOFFE (EMM) to
advance pointers to the next field in the format table and store the data
type of the next field in the ATT. Then VMM1.

(VMM1). Clear the character count in this field (ATT/FCNT).

(VMM2). Check the next field for verification type. If no verify, update
pointers to end of field and jump to EORTST. If sight verify, jump to
SIGHT. If other verify, jump to SETFLD. ,

(SETBEG). Repositions the character count in the current record and field
to the beginning of the next field.

(CHKVRY). Check the ATT to see if verification by format is requested.
If so, check the verification type of the current field. If the field is
specified for verify, mask off bit 0 to show 0 = sight verify, 1 = key
verify. In addition, if the field is autodup or autoskip, is not the first
record in the format, or has the autodup change flag set, then skip
this field. Otherwise, verify it.

If verification by format is not requested or the field is not specified
for verify, check for verification of error flagged fields or unbalanced
counter fields.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME____VMM - Verify Mode Processor_____Level 6

INTERFACE

1. ENTRY METHOD AND INPUT PARAMETERS

    In all cases, 1 = current ATT base address.
    VMM: Direct jump.
           ATT/CURCH = current character entered in Verify mode.
    GETNZ: As VMM.
    MATCH: Direct jump.
           Character entered in Verify mode is same as character entered in
           Entry mode.
    ENDOF1: Direct jump.
    NOVAL: Direct jump
    RVFVER: Direct jump
    EORTST: Direct jump
    VMM1:   Direct jump
    VMM2:   Direct jump
    SETBEG: Direct jump
    CHKVRY: Direct jump

2. EXIT METHOD AND OUTPUT PARAMETERS

    In all cases, 1 = current ATT base address.
    VMM: Jump to MATCH -- if character matches
            Jump to ERROR -- if character mismatch or sign missing
    GETNZ: As VMM
    MATCH: Jump to CLRSR -- if not at end of field
            Jump to EMMSKP -- if forced skip at end of field or fill characters
                              were bypassed.
            Jump to ENDOF1 -- if fixed keying or no sign entered
    ENDOF1: Same as RVFVER
    NOVAL: Same as RVFVER
    RVFVER: Jump to CLRSR
    EORTST: Jump to EORVR
    VMM1:
    VMM2:   Jump to CLRSR

    SETBEG: P + 1 return to caller
            ATT/RCNT -- updated to next field
            ATT/FCNT/0-7 = 0

    CHKVRY: P + 1 Return to caller
            A   0 No verify
            A = 0 Sight verify
            A = 1 Key verify by format
            A = 2 Error flagged field requiring key verify
            A = 3 Unbalanced counter field requiring key verify.

CONTROL DATA CORPORATION ● PSSD ● SOFTWARE DOCUMENT

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME___VMM - Verify Mode Processor_____level 6

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LOCORE

        LPMASK (2)
        NZERO ($12)
        ONEBIT ($23)
        ZROBIT ($33)
        SST ($47)
        CLRSR ($93)
        GBYTE ($94)
        ERROR ($96)
        DISPLY ($A0)

    b. ATT

        CMODE (2)
        CURCH (2)
        FCNT (3)
        RCNT (4)
        MISCH (5)
        FDTP (13)
        SW1 (19)
        SW2 (20)
        SWITCH (24)
        RCD (32)
        COUNT (36)
        ATEMP (73)

    c. SST

        CFFA (12)
        STEMP (20)

    d. ERROR CODES

        ERR2 (2)   Sign missing
        ERR5 (5)   Mismatch

2. MACROS USED

   None

AA5641   PRINTED IN THE USA

3. PROGRAM DEPENDENCIES

    EMM-   VALDTE
            BOFFE
            TSTREC
            EMMSKP

    DISPLY- DISPLY
            DCAROL
            DEOF
            DBIF
            DEOREV
            DBSF
            DSPFLD
            DCOL
    CLEARS- CLRFLD
    CREGEN- REGEN
    EORVR
    FMTPAR
    GSE-   GBYTE
            ERROR
    ENEXSR- CLRSR

4. DATA AREAS USED

    ATT
    SST

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $135_{16} = 309_{10}$ words.

2. RELOCATION TYPE

    Loader relocatable.

3. ENTRANCY CLASSIFICATION

    Dormant reentrant.

AA5641

DOCUMENT CLASS ___CYBERDATA IMS___

PROGRAM NAME___WBT - Write Batch to Tape___   Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    WBT is the first overlay of three overlays function, which transfers
    batches from disk to tape.

    The command format is as follows:

    WBT, job [,start batch] [,finish batch] [,conditions]   ,
        [,STOP]   [,NEW]   [,MTF = n]

    "job" is the job name of the job to be written. "Start batch" is the
    number of the first batch to be written to the tape by the command
    (defalut is 1). "Finish batch" is the number of the last batch to be
    written to the tape by the command (default is 999). "condition is
    one or more of the following key parameters"

$$WR = \begin{Bmatrix} Y \\ N \\ B \end{Bmatrix} \quad , \quad UB = \begin{Bmatrix} Y \\ N \\ B \end{Bmatrix} \quad , \quad ER = \begin{Bmatrix} Y \\ N \\ B \end{Bmatrix}$$

    Y - represents YES, N - represents - NO and B represents both (default is N).

    WR = Y means that only "written" batches will be written to the tape.

    WR = N means that only unwritten batches will be written to the tape.

    WR = B means that batches will be written to the tape, regardless of
    their "written" status.

    UB and ER relate to "unbalanced" and "erroneous" statuses, in the same
    way.

    STOP - signifies that if one of the batches, in the range start batch -
    finish batch, is not completed or does not meet the above mentioned
    conditions, than the function will be terminated.

    NEW - specifies that a new file will be opened by the command, regardless
    of the "batch grouping factor" parameter in the Mag. Tape Format.

    MTF = n specifies that MTF number n will be used in this command, instead
    of the job MTF.

    The main functions of the WBT program are:

AA5641   PRINTED IN THE USA

a. Check the command parameters and put them in the Mag. Tape Communication Area (MTCA). An erroneous parameter terminates the command.

b. Get a buffer for the command execution.

c. Get the appropriate MTF.

d. Open the tape, i.e. search for the Cyberdata tape trailer, check its fields, and position the tape to the location of the next record to be written.

e. Get tape recovery record from the "Tape Recovery Area", on the disk.

f. Load code conversion program, if needed.

g. Call MATER.

2. DETAILED DESCRIPTION

On entering the WBT program, the command parameters are stored in the Interrogate buffer in the form entered by the operator.

The program performs the following steps: -

a. Check job name and store it in JNAME.

   If job name was not keyed or is too long, theprogram is terminated. (later the job name is checked against the LJD).

b. Check and store the start and finish batch numbers. If start batch was not keyed put 1; if finish batch was not keyed put 999.

c. Check the key-word parameters: WR = , UB = , ER = , STOP, NEW, MTF = (regardless of their order in the Int. buffer). According to the key-word the program goes to the relevant routines, to check and store the parameters.

   The WR, UB, ER, STOP and NEW Parameters are stored as bits of the STATUS word, in MTCA. The MTF number, is stored in MTFNUM. If one of the parameters is erroneous or appears more than once, the program is terminated.

d. Check the job name (TNAME) against the LTD. If it does not match - terminate.

e. Get the MTF number from MTFNUM, or if it does not appear, from the job entry in the LTD, and read the MTF.

f. Check if recording code, in the MTF, is acceptable by the mag. tape (BCD is not accepted by 7 tracks tape).

   If the recording code was not entered in the MTF, put EBCDIC for 9-track tape and BCD for 7-track tape.

g. For 7-track tape, compare the block length, in the MTF to the driver buffer length. The second must be greater or equal.

h. Get the following buffers:

- EOD buffer (connected to EODADR) for the EOD record and for the format number tables.

- Output buffer (connected to OUTBUF) for the output blocks which are written to the tape.

- TRB buffer (connected to TBB) for a data record.

- Input buffer (connected to INBUF) for the input blocks which are read from the data disks.

i. Search for the EOD1 record on the tape. The program skips 4 records back backwards on the tape, and checks if the next two records are Tape Mark and EOD1 record; if not the tape is rewound and the searching starts from the beginning. The search for EOD1 continues until the EOD1 record is found or until end of tape, or read error (including short read) occurs.

j. Read the EOD record (the record following the EOD1 record).

k. Compare the job name of the command with the job name of the EOD record; if not equal, print a message to the operator and receive "CU" to continue or "TR" to terminate.

l. Get recovery record area from the recovery area on the disk. If the recovery area is full, the command execution is terminated.

m. If the recording code is not ASCII or BCD, then load the appropriate code conversion routine, which resides in the memory during the command execution.

n. Convert the Padding character, in the MTF, according to the recording code.

o. Release WBT and call in MATER.

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___WBT - Write Batch to Tape_____Level 6

INTERFACE

1. ENTRY METHOD

   WBT receives control from DSKSUP via ENTPRG.

2. INPUT PARAMETERS

   a. Command parameters - received in the Interrogate buffer, in the form
      entered by the operator.

   b. Mag. Tape Format (MTF) - read from the MTF's table (start address is in
      MTFDA of EST). See "MAG TAPE FORMAT TABLE" in CYBERDATA SOFTWARE
      USER'S GUIDE.

   c. Cyberdata tape trailer - this trailer is composed of 3 records,
      proceeded by a Tape Mark (EOF). The records are:

           EOD1 record, EOD record and the Noise record.

      The EOD1 is a 40 words record in which the first two words contain the
      characters EOD1, written in the tape recording code. The content of
      the other 38 words is not important.

      The EOD record is also 40 words long, written in binary mode, immediately
      after the EOD1 record. See "END OF DATA TRAILER" in CYBERDATA SOFTWARE
      USER'S GUIDE.

      The noise record is a short block of 3 words. It signals the end of
      data when the tape was recorded in a code which differs from the MTF
      recording code.

3. EXIT METHOD

   a. Normal exit is to MATER, by RTJ - (ENTPRG).

   b. Error exit is to REJSUP, by JMP - (REJSUP).

4. OUTPUT PARAMETERS

   a. MTCA - see MAG TAPE COMMUNICATION AREA, in CYBERDATA SOFTWARE USER'S
      GUIDE.

   b. The EOD buffer, with the EOD RECORD.

c. The following messages may be printed by the WBT module:

1) INVALID JOB NAM

2) INVALID BATCH NUMBER

3) INVALID PARAMETER

4) MEMORY FULL - ENTRY

5) INCOMPLETE COMMAND

6) TAPE I/O ERROR

7) END OF TAPE

8) ANOTHER JOB RECORDED ON THIS TAPE

    TYPE CU/TR TO CONTINUE/TERMINATE.

9) INVALID MTF NUMBER OF MTF PARAMETER

10) BLOCK TOO LONG FOR 7-TRACK TAPE

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. SST

        TEMP1, TEMP2

    b. TRANSFER VECTOR

        ENTSR
        RTNSR
        GETBIN
        TAPIO
        GETALP
        ENTPRG
        EXDFUN
        SUPRW
        GETCOM
        CAM
        WAIT
        REJSUP
        EXTSR
        CLCDWA
        REDDWA
        WRTDWA
        RELFOR

   c. EST

      MTFDA - MTF disk address (EST word 42).

   d. ATT

      JIDX - job index
      MAXREC - job maximum record length
      INBUF - input buffer pointer
      TRB - record buffer
      TRMNT - terminate flag
      IRB - interrogate buffer pointer
      OUTBUF - tape output buffer pointer
      BATCH - length number and disk L.U.N.
      ATEMP - ATT temporary word

   e. COMMUNICATION AREA

      SVCA - start of communication area
      BCOUNT - tape I/O blocks counter
      MTCA - start of Mag. Tape Communication Area
      MTFENT - mag. tape routines entry parameter
      MTFEXT - mag. tape routine exit program index
      EODADR - EOD buffer pointer
      MTF - Mag. Tape Format (5 words)
      STRBAX - start batch number
      ENDBAX - finish batch number
      STATUS - status conditions word
      TRANUM - Tape Recovery Area Number
      JNAME - job name (3 words)
      MTFNUM - MTF number

      <u>Note:</u> see MTCA table in Cyberdata software user's guide.

   f. GENERAL EQUATES

      LOW CORE EQUATES
      ASCII character equates for: N, X, Y, = ,0
                                     P, B, F, W
      FR - ASCII format read PIO request code
      FW - ASCII format write PIO request code
      BFR - Binary format read PIO request code
      CODFR - Code conversion format read PIO request code
      MO - Motion PIO request code
      MAXTRA - Maximum number of tape recovery areas.
      EODLNG - EOD record length
      FMTLNG - Format numbers table length
      MATER - MATER program index

2. MACROS USED

   None.

3. PROGRAM DEPENDENCIES

   a. ENTRY POINTS

      None.

   b. EXTERNALS

      SETMOV
      GETLTD
      PRTMSG
      BD
      LOG1A - LOG1A address
      SITRDA - Tape Recovery Disk Address
      CODPRG - Code Conversion Program indices table (in PIO)

   Transfer vector routines - see 1, EQUATES, in this chapter.

4. DATA AREAS AND SYSTEM TABLES USED

   The following system tables are used in WBT: SST, EST, ATT, MTF, Tape
   Recovery Area, LOG1A.

   Buffers allocated or used by WBT are:

   IRB - Interrogate buffer
   INBUF - Input buffer
   OUTBUF - Output buffer
   TRB - Record buffer
   EOD - EOD record and Formats number table buffer.

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

   830 words.

2. RELOCATION TYPE

   Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

   Reentrant, partially locked.

AA5641   PRINTED IN THE USA

DOCUMENT CLASS ___WRTFMT_____

PROGRAM NAME_____Write Format to Disk_____

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

   WRTFMT writes a format description on the disk. If a previous format exists
   with same format number, the new format description is written in the disk
   space occupied by the previous format description, unless there is not sufficient
   space for the new format. In that case, available disk space is used.

   Format Link Table (FLK) and Legal Format Directory (LFD) are updated.

2. DETAILED DESCRIPTION

   a. (LOCKED). If (LOCK) ≠ 0 (WRTFMT module is busy), call WAIT to delay one
      processing cycle before repeating this step.

      o If (LOCK) = 0, set it to 1.
      o Set (NFL) = New Format String Length.
      o Set (NRL) = Maximum no. of Characters in Record - 1 for New Record.
      o Set (NFLS) = New Format String Length in Sectors.
      o Set (FMTN) = Format Number.
      o Call GFCORE to bring previous format definition into core.
      o If format does not exist, go to step b.
      o Set (OFL) = Previous Format String Length.
      o If (NRL) ≠ max. no. of characters in Record - 1 for previous format, go
         to step e with (Q) = 1 (Record Length Error).
      o If Previous Format occupies fewer sectors than new Format, go to step b.
      o Call CONMMA to compute disk address for previous format in Legal Format
         Directory (LFD).
      o Call RW2WRD to read entry from LFD.
      o Go to step d.

   b. (NEWFMT). Call CONNMA to calculate disk address of Format Link Table (FLK)
      header.

      o Call RW2WRD to read 2-word FLK header.
      o If there are sufficient available sectors on the current track for new
         format description, go to step c.
      o If FLK table is full, go to step e with (Q) = 2 (FLK Full).
      o Set (AVLSCT) = 16 (Full track available).
      o Call GITRT to allocate an available track.
      o If no track is available for assignment, go to step e.
         with (Q) = 3 (Disk Full).
      o Set (TRK) = address of allocated track.
      o Call RW2WRD to write 2-word FLK header showing full track available.
      o Call CONMMA to calculate disk address of current FLK entry.
      o Call RW2WRD to write new format track address in current FLK entry.
      o Repeat step b.

228

DOCUMENT CLASS ___WRTFMT___

PROGRAM NAME___Write Format to Disk___

    c. (WRT1). Set (PAVLS) = Number of available sectors on current track.

- Set (AVLSCT) = Number of available sectors - number of sectors in new format.

- Call RW2WRD to rewrite FLK header with new value of AVLSCT.

- Call COMMA to calculate address of current FLK entry.

- Call RW2WRD to read current FLK entry.

- Set (SECTOR) = sector address of available area.

- Call COMMA to calculate disk address of format entry in LFD.

    d. (WFMT). Call RW2WRD to write LFD entry with new format sector address and new format length.

- Call WRTSEC to write the format description string to disk.

- Set (Q) = 0.

    e. (ERROR). Save (Q) in ATT/ATEMP.

- Call RELFOR to release previous format description from core.

- Call EXDFUN to release WRTFMT module from core.

- Set (LOCK) = 0 (WRTFMT module not busy).

- Set ATT/PFBA = 0 (No format).

- Set ATT/FMTNO = 0 (No format).

- If ATT/CALRPI = 0 (Caller's Program Index), go to step f.

- Call ENTPRG to load and execute entry point 1 in calling module.

    f. (NORTN). Call RELFOR to release new format description from core.

- If ATT/ATEMP = 0 (No errors), exit to SV.

- Convert error code in ATT/ATEMP from 1, 2, or 3 to 10, 50, or 51, respectively, and exit to REJSUP to issue one of the error messages:
  INVALID RECORD LENGTH   (10)
  FORMAT LINK TABLE FULL (50)
  DISK FULL   (51)

INTERFACE

1. ENTRY METHOD

    WRTFMT is disk-resident. It is called via ENTPRG routine. Used as a second overlay of EFM and of LODFMT.

2. INPUT PARAMETERS OR CONDITIONS

    (I) = Base address of Active Terminal Table (ATT).
    CALPI = Function index of caller in case return is required.
    FMTBAS = Pointer to the new format in core.

DOCUMENT CLASS ___WRTFMT___

PROGRAM NAME___ Write Format to Disk ___

3. EXIT METHOD

Exit is to REJSUP if an error is detected, to the second entry point of the disk resident calling program if it was indicated.

4. OUTPUT PARAMETERS OR CONDITIONS

(I) = Base address of ATT.

ASSEMBLY CHARACTERISTICS

1. EQUATES

a. LOCORE

LPMASK ($2)
NZERO ($12)
ZERO ($22)
ONEBIT ($23)
ZROBIT ($33)
CLCDWA ($90)
EXTSR ($92)
GFCORE ($9D)
ENTPRG ($9E)
EXDFUN ($9F)
WRTSEC ($A3)
REDWRD ($A9)
WRTWRD ($AA)
RELFOR ($AD)
WAIT ($B0)
SV ($B1)
REJSUP (B2)

b. ATT

PFBA (07)
FMTNO (13)
PI (21)
SUPCOM (36)
CALRPI (SUPCOM + 6)
FMTBAS (SUPCOM + 10)
ATEMP (73)

c. EXTERNALS

SIFLNK   - FLK disk address, defined at SI time.
SILFD   - LFD disk address, defined at SI time.
G1TRT   - Get one track, entry point in DAM.

DOCUMENT CLASS __WRTFMT__

PROGRAM NAME __Write Format to Disk__

2. MACROS USED

None

3. PROGRAM DEPENDENCIES

PIO	CLCDWA	Calculate Disk Word Address
	WRTSEC	Write Sector
	REDWRD	Read Disk Word Address
	WRTWRD	Write Disk Word Address
ENEXSR	EXTSR	Exit subroutine
	WAIT	One-cycle delay
ENTPRG	ENTPRG	Enter a Disk Resident Program
	EXDFUN	Release a Disk Resident Program
GFCORE	GFCORE	Get Format into Core
	RELFOR	Release format
SUPER	SV	Supervisor Command Processor
	REJSUP	Reject Supervisor Command
DAM	GITRT	Allocate a disk track.

4. DATA AREAS USED

ATT   Active Terminal Table

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

$BE_{16} = 190_{10}$ words

2. RELOCATION TYPE

Disk resident, run anywhere.

3. ENTRANCY CLASSIFICATION

Not reentrant.

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME_____XPL_____ Level 6

PROGRAM FUNCTION

1. GENERAL DESCRIPTION

    XPL is a supervisor function to display the layout of any supervisor command
    or a list of expanded names of the supervisor commands where-

    a. XPL   display a list to expanded names of the supervisor commands.

    b. XPL,aaa   display the layout of aaa supervisor command.

2. DETAILED DESCRIPTION

    XPL first enters SETMØV to enable program moving, checks the 4th character
    keyed to be a comma or end of record, else go to REJSUP to indicate invalid
    command format (ERR9). If a comma is found, collect 3 characters of command
    mnemonic, check them to be valid ASCII characters and pack them into one
    word key. Check if end of record follows, else go to ERR9. Then a look up
    loop is performed to find a corresponding entry in a keys table. If not found
    go to REJSUP to indicate that program does not exist (NØPRG). Else the
    distance from beginning of table of that entry is saved and a core request
    for lengths table of layout texts is issued (the lengths table and layout
    tests reside in disk as a miscellaneous program). Then the table disk
    address is calculated and the table is brought by REDTAB subroutine. If
    names list is requested (ATT/FLAG=FFFF) another 25 word core request is
    issued. But if command layout is requested, the text length of the
    requested command is extracted from the lengths table, the relative
    location of the length table area is released and a new core request for
    the text is issued.

    Then according to the calculated disk address a command layout text or an
    expanded name is read by REDTAB subroutine. If it is first time an
    interrogate buffer of 48 characters is requested and if the list device
    is not TTY a page eject request is issued. Then blanks are spread in
    the interrogate buffer and the pointers of the input/output buffers are
    initiated. If a names list is requested the command mnemonic is stored
    at the beginning of the output buffer and then the required text is moved.
    Output request to display the line is issued and if CANCLE/SX was keyed,
    exit to SV after releasing the requested core (EXIT). If function is
    not terminated and list device is 970-480 and end of screen occurred,
    enter CØNCAN to check cancle/continue. If cancel go to EXIT, else
    perform top screen request for 970-480 device then spread blanks in output
    buffer. If names list is requested and last command name displayed go to
    EXIT, else loop back to read next expanded command name (NEXT3). If
    command layout is requested and it is not last line of text loop back to
    display next line (NXTLIN) else go to EXIT.

INTERFACE

1. ENTRY METHOD

    XPL is a supervisor function loaded by ENTPRG.

    Only entry point is XPL.

2. INPUT PARAMETERS OR CONDITIONS

    1 = current ATT address

    Operator enter either

          XPL
          XPL,aaa

3. EXIT METHOD

    a. Jump to REJSUP with appropriate error codes

    b. Jump to SV

4. OUTPUT PARAMETERS OR CONDITIONS

    1 = current ATT address

ASSEMBLY CHARACTERISTICS

1. EQUATES

    a. LØCØRE

        ZERØ ($22)
        LPMASK (2)
        ØNEBIT ($23)
        SST ($47)

    b. SST

        PRD (3)
        STEMP (20)

    c. TRANSFER VECTØR

        CLCDWA ($90)
        ENTSR ($91)
        EXTSR ($92)
        SUPRW ($97)
        GETALP ($9B)
        GETCØM ($9C)
        REDWA ($A9)
        CAM ($AE)
        SV ($B1)
        REJSUP ($B2)

DOCUMENT CLASS __CYBERDATA IMS__

PROGRAM NAME __XPL__   Level 6

    d. ATT

        PINT (27)
        CANC (23)
        SCA (36)
        SCØ (SCA+1)
        FUNAME (SCA+10)
        NAMLIM (SCA+11)
        ITMLNG (SCA+12)
        CØRADR (SCA+13)
        LUTYPE (SCA+14)
        INTIX (SCA+15)
        FLAG (SCA+16)
        LTADD (SCA+17)
        FUNCNT (SCA+18)

    e. CONSTANTS

        A   ($41)
        B   ($42)
        C   ($43)
        D   ($44)
        E   ($45)
        F   ($46)
        G   ($47)
        H   ($48)
        II ($49)
        J   ($4A)
        K   ($4B)
        L   ($4C)
        M   ($4D)
        N   ($4E)
        O   ($4F)
        P   ($50)
        Q   ($51)
        R   ($52)
        S   ($53)
        T   ($54)
        U   ($55)
        V   ($56)
        W   ($57)
        X   ($58)
        Y   ($59)
        Z   ($5A)
        FW (3)   Formatted write code
        MØT (5)   Motion request code

DOCUMENT CLASS ___CYBERDATA IMS_____

PROGRAM NAME___XPL_____ Level 6

2. PROGRAM DEPENDENCIES

PIO	- (SUPRW)	- Read from/write to supervisor devices.
	(CLCDWA)	- Calculate word address of disk.
	(REDWA)	- Read n words from disk according to word address.
	(GETBUF)	- Get interrogate buffer for terminal.
CØNCAN	-	- Wait for continue or cancel key.
SCANV	- (GETALP)	- Get alpha character from input string.
	(GETCØM)	- Get next comma or end of record from input string.
CAM	-	- Allocate core buffer.
SUPER	- (SV)	- Conclude supervisory function request.
	(REJSUP)	- Generate error message to supervisor.
ENEXSR	- (ENTSR)	- Common subroutine entry.
	(EXTSR)	- Common subroutine exit.
	(SETMØV)	- Enable disk resident program to move.

3. DATA AREAS USED

    SST
    ATT

PHYSICAL CHARACTERISTICS

1. MEMORY SPACE REQUIREMENTS

    $13F_{16} = 319_{10}$ words.
    Buffer of 25_{10} words for text display.

2. RELØCATION TYPE

    Run anywhere.

3. ENTRANCY CLASSIFICATION

    Reentrant.