# IBM Technical Newsletter

Base Publ. No. GY20-0591-1

This Newsletter No. GN20-2503

Date January 24, 1972

Previous Newsletter Nos. None

**Control Program-67/Cambridge Monitor System**
**(CP-67/CMS) Version 3.1**
**CMS Program Logic Manual**
**Program Number 360D-05.2005**

This Technical Newsletter, a part of Version 3, Modification Level 1, of Control Program-67/Cambridge Monitor System, provides replacement pages for the subject manual. These replacement pages remain in effect for subsequent versions and modifications unless specifically altered. Pages to be inserted and/or removed are listed below.

Pages

3,4
171-176
283, 284

Minor additions and changes have been made to provide program support information on the IBM 3420 Magnetic Tape Unit.

A vertical rule in the left margin indicates that a change has been made to either text or illustration.

Please file this cover letter at the back of the manual to provide a record of changes

*IBM Corporation, Technical Publications Dept., 1133 Westchester Avenue, White Plains, N. Y. 10604*

Printed in U.S.A.

utilization. He can also rename any command and define his own abbreviations to be used.

## Library Facilities

CMS provides library facilities for program libraries. The user can generate his own libraries or add, delete, or list entries in existing libraries. He can also specify which libraries to use for program assemblies as well as program execution.

## Page Release Facility

Certain CMS routines include a page release facility. This means that following a successful completion and before returning to the user or caller, the routine references NUCON and turns a page release flag on. When the routine then returns to INIT, INIT checks this flag. If it is on, INIT issues a diagnose X'10' to CP to release user pages from X'12000' up to the value in LOWEXT.

For the user to prevent this release of pages, the CMS VSET RELPAG OFF command should be issued. The commands that have the page release facility are: ASSEMBLE, CEDIT, COMBINE, COMPARE, EDIT, FORTRAN, MACLIB, MAPPRT, PLI, SORT, SPLIT, TAPE, TXTLIB, and UPDATE.

## CMS Core Requirements

CMS has a prerequisite for 80K bytes of virtual memory for the nucleus, transient area, and loader tables. At login time, core space for user file directories is allocated dynamically as required. The rest of core storage is available to user programs.

## CMS Batch Monitor

As well as being a conversational monitor, CMS provides a batch facility for running CMS jobs. The CMS batch monitor accepts a job stream from a tape unit or from the card-reader and writes the output either on tapes, the printer, or the card-punch. The job stream can consist of a System/360 Operating System SYSIN job stream with FORTRAN (G), and Assembler (F) compile, load, and go jobs or it can consist of CMS commands along with control cards and card decks for compile, load, and go jobs for all the CMS supported compilers.

Just as the conversational CMS does, the batch monitor can run from either a virtual machine or a real machine. Under CP, it can be used as a background monitor along with other conversational CMS users.

To eliminate the possibility of one job modifying the CMS batch monitor's nucleus in such a way as to affect the next job, the batch monitor is re-IPLed before each job begins. Files can also be written onto the batch monitor's primary disk and then punched or printed, such as files written by FORTRAN programs; these files should be of limited size and considered as temporary, as they are erased at the completion of each job.

## MACHINE CONFIGURATION

Whether running on a real (see Note below) or a virtual machine, CMS expects the following machine configuration:

| Device | Virtual Address | Symbolic Name | |
|---|---|---|---|
| 1052 | 009 | CON1 | console |
| 2311,2314 | 190 | DSK1 | system disk (read-only) |
| 2311,2314 | 191** | DSK2 | primary disk (user files) |
| *2311,2314 | 192** | DSK3 | temporary disk (work space) |
| *2311,2314 | 000** | DSK4 | A disk (user files) |
| *2311,2314 | 000** | DSK5 | B disk (user files) |
| *2311,2314 | 19C** | DSK6 | C disk (user files) |
| 1403 | 00E | PRN1 | line printer |
| 2540 | 00C | RDR1 | card reader |
| 2540 | 00D | PCH1 | card punch |
| *2400, 3420 | 180 | TAP1 | tape drive |
| *2400, 3420 | 181 | TAP2 | tape drive |

at least 256K bytes of core storage, 360/40 and up

*The 2311 or 2314 for the temporary disk, the A, B and C disks, and the two tape drives are optional devices; they are not included in the minimum configuration.

**The specified virtual addresses may be changed at any time by the CMS LOGIN command.

Note: For use on a real machine not having this I/O configuration, the device addresses can be redefined at 'load' time.

Under CP, of course, these devices are simulated and mapped to different addresses and/or different devices. For instance, CMS expects a 1052 printer-keyboard operator's console, but most remote terminals are 2741's; CP handles all channel program modifications necessary for this simulation.

CMS allows the user to add his own programs for I/O devices not supported by the standard system. CMS also provides for dynamic specification of SVC routines.

The system disk, located at address 190, is read-only and contains the CMS system commands. These system programs are physically divided into two groups: nucleus functions and disk-resident command modules. The nucleus programs are loaded into main storage during initial program load (IPL) and remain resident throughout system operation. The disk resident modules are loaded into main storage only when their services are needed. Certain disk resident programs are loaded into the transient area. The primary disk, 191, is a read-write disk and normally is the first user disk. Files that the user wishes to retain for use across terminal sessions are stored on one of the user's disks. Information stored on the primary disk remains there until it is deliber-

If a '?' is entered as the only parameter, a brief status of all currently logged-in disks is typed. This brief status gives disk labels, disk address, disk mode, and R/O if the disk is read-only.

If an additional parameter of '?' is given, added information is typed, specifically whether the disk is 2311 or 2314, and whether it is read-only or read-write.

If a user has been logging in several disks and wants to be sure which ones he has logged in at the moment, a call to 'STAT?' will provide the clue needed as to which disks are indeed logged in. (This would also show the order of search.)

The STAT command is transient-disk-resident. The name of the text deck is 'STATDSK'; thus a procedure for generating a new module of 'STAT' would be as follows (or equivalent):

```
LOAD    STATDSK   (TRANS TYPE
GENMOD  STAT
```

The logic of STAT is simple. ADTLKP or ADTNXT is called to find the appropriate Active Disk Table block; if the flag bits indicate the disk is logged in the pertinent disk, statistics are simply converted to printable form and typed. Leading zeros are eliminated by shifting the typeout left for any leading zeros found, and adding a blank at the end, with the subsequent call to TYPLIN deleting the trailing blanks.

## TAPE

FUNCTION: To dump disk files to tape, to restore files that were dumped to tape back onto disk, to rewind a tape, to write an end-of-file mark on a tape, or to skip to the next end-of-file mark on a tape.

ATTRIBUTES: Disk resident

CALLING SEQUENCE:

```
LA    1, PLIST
SVC   X'CA'
```

*REWIND:*

```
PLIST DC    CL8'TAPE'
      DC    CL8'REWIND'
      DC    CL8'TAPn'       CMS device name (optional)
```

WRITEOF:

```
PLIST DC    CL8'TAPE'
      DC    CL8'WRITEOF'
      DC    CL8'n'          Write 'n' EOF marks (optional)
      DC    CL8'TAPn'       CMS device name (optional)
      DC    CL8'800BPI'     or '1600BPI' tape density (optional)
```

SKIP:

```
PLIST DC    CL8'TAPE'
      DC    CL8'SKIP'
      DC    CL8'n'          number of EOF marks (optional)
      DC    CL8'TAPn'       CMS device name (optional)
```

DUMP:

```
PLIST DC    CL8'TAPE'
      DC    CL8'DUMP'
      DC    CL8'filename' or CL8'*'
      DC    CL8'filetype' of CL8'*'
      DC    CL8'filemode'   default of 'P1' if omitted
      DC    CL8'TAPn'       CMS device name (optional)
      DC    CL8'800BPI'     or '1600BPI' tape density (optional)
```

LOAD:

```
PLIST DC    CL8'TAPE'
      DC    CL8'LOAD'
      DC    CL8'n'          stop after 'n' EOF marks (optional)
      DC    CL8'TAPn'       CMS device name (optional)
```

SCAN:

```
PLIST DC    CL8'TAPE'
      DC    CL8'SCAN'
      DC    CL8'n'          stop after 'n' EOF marks (optional)
      DC    CL8'TAPn'       CMS device name (optional)
```

SLOAD:

```
PLIST DC    CL8'TAPE'
      DC    CL8'SLOAD'
      DC    CL8'filename' or CL8'*'
      DC    CL8'filetype' or CL8'*'
      DC    CL8'n'          stop after 'n' EOF marks (optional)
      DC    CL8'TAPn'       CMS device name (optional)
```

Note: The default value for 'TAPn' is TAP2' and the default value for 'n' is'1'.

OPERATION: The operation of the TAPE command program depends on whether the calling program specifies REWIND, WRITEOF, SKIP, DUMP, LOAD, SCAN, or SLOAD.

REWIND: TAPE calls the TAPEIO function program to rewind the tape. It then returns (via SVCINT) to the calling program, which is usually INIT.

WRITEOF: TAPE calls the TAPEIO function program to write an end-of-file marker. It then returns to the calling program. If a tape density is specified, the corresponding modeset code is placed in the TAPEIO parameter list.

SKIP: TAPE repeatedly calls the TAPEIO function program to read successive records from the tape until an end-of-file marker is encountered. It then returns to the calling program.

DUMP: TAPE calls the FSTLKP function program to locate the file status table (FST) block for the file. TAPE then temporarily alters the FST block characteristics to 800-byte fixed-length records. TAPE then calls the RDBUF function program to read the first 800-byte block in the file into a buffer and the TAPEIO function program to write the data block from the buffer onto tape. If a tape density is specified, the corresponding modeset code is placed in the TAPEIO parameter list. TAPE repeats this procedure of calling RDBUF and TAPEIO for each data block in the file. When an end-of-file is reached, TAPE calls TAPEIO and writes a trailer record on the tape. The trailer record identifies the file and contains an N in the fifth byte, the last 20 bytes of the file, status table for the file in bytes 6 - 25, and the file designation in bytes 70 - 87. Finally, TAPE restores the FST block and calls the FINIS command program to close the file. It then returns to the calling program.

Note: Each data block is written, as a single tape record. The tape record is 805 bytes long. The first four bytes contain a code indicating that the record was produced by the TAPE command program. The next byte is zero, except for the trailer record. The remaining 800 bytes contain the data block.

LOAD: TAPE calls the ERASE command program to erase the file (if any) designated as (DISK) (TFILE) P3. Next it calls the TAPEIO function program to read the first record on the tape. (If the record was produced by the dump portion of TAPE, it will be 805 bytes in length and contain a code in the first four bytes and a data block in the last 800 bytes). TAPE then checks the code in the record. If invalid, TAPE issues a message to the effect that the tape is not in tape load format,and returns to the calling

program. If the code is valid, TAPE determines if it is a trailer record (that is, one with N in the fifth byte). If not a trailer record, TAPE calls the WRBUF function program to write the data block contained in the record into a file designated as (DISK) (TFILE) P3. Then TAPE calls the TAPEIO function program to read the next record from the tape, which it processes in the same manner.

If the tape record read is a trailer record, TAPE calls the FINIS command program to close the disk file ((DISK) (TFILE) P3) created from the first tape file. It then calls the ERASE command program to erase the file (if any) that has the same designation as the tape file just converted to a disk file. Next, TAPE calls the FSTLKP function program to locate the file status table for the disk file just created. (This is again (DISK) (TFILE) P3.) Subsequently, TAPE overlays the last 20 bytes of the file status table, (except for the first chain link address) with the corresponding data from the trailer record and moves the filename, filetype, and date last updated for the tape file from the trailer record into the corresponding locations in the FST block. The directory is then updated with a call to UPDISK. This completes the conversion of the first file on the tape to a disk file, and TAPE calls the TYPLIN function program to type a message at the terminal to the effect that the file has been loaded. TAPE processes the next file on the tape in a similar manner. When the 'n'th end-of-file on the tape is encountered, TAPE returns to the calling program. If 'n' was not specified, TAPE returns to the calling program when the first end-of-file is reached.

SCAN: TAPE sets SCANSWT to disable disk operation and to enable end-of-file printout and then branches into the code for TAPE LOAD. The effect of this command is to list the contents of a tape (including end-of-file marks) at the terminal until the 'n'th end-of-file mark is encountered. The default value of 'n' is one.

SLOAD: TAPE searches for the file whose filename and filetype were specified in the parameter list. When the matching file is found, SLDSWT is set, and TAPE branches into the TAPE LOAD code to copy the file from tape to disk. SLDSWT enables control to return to the SLOAD coding after the file is copied. If neither the filename nor filetype was '*' TAPE returns control to the calling program, otherwise the above search and load procedure is continued until the 'n'th end-of-file mark is encountered. Control is returned to the user when the 'n'th end-of-file mark is encountered even if no files were copied from tape to disk. The default value of 'n' is one.

## TAPEIO

FUNCTION: To (1) read or write tape records, (2) rewind the tape, and (3) write an end-of-file marker on tape.

ATTRIBUTES: Disk resident, transient

Note: For a detailed explanation on TAPEIO, see the write-up on the TAPEIO function.

## TAPRINT

FUNCTION: To print the contents of a tape containing assembler or FORTRAN LISTING files.

ATTRIBUTES: Disk resident

CALLING SEQUENCE:

```
        LA      1, PLIST
        SVC     X'CA'
        .
        .
        .
PLIST   DC      CL8'TAPRINT'
       [DC      CL8'          ']      symbolic tape name
```

OPERATION: TAPRINT receives the symbolic name of the tape to be printed from the parameter list and inserts it into the calling sequence to the TAPEIO function program. It then calls the TAPEIO function program to read the first record from the tape. This record contains 10 blocked print line images. Next, TAPRINT repeatedly calls the PRINTR function program to print each of the 10 print line images on the printer. TAPRINT repeats this process of reading a record from tape and printing the 10 print line images contained therein until it encounters an end-of-file on the tape. At this time, it calls the TYPLIN function program to type a message at the terminal signaling the end-of-file. It then prints the next file on the tape in a similar manner. When two consecutive end-of-files are encountered, meaning that the end of the tape has been reached, TAPRINT calls the TYPLIN function program to indicate this. It then calls the TAPEIO function program to rewind the tape, calls the CLOSIO command program to close printer operations, and returns to the calling program.

Note: If the calling program does not provide a symbolic tape name, TAP2 is assumed.

## TPCOPY

FUNCTION: To copy tape files.

ATTRIBUTES: Disk resident.

CALLING SEQUENCE:

```
        LA    1, PLIST
        SVC   202
        .
        .
        .
PLIST   DC    CL8'TPCOPY'
        DC    CL8'TAP1'
        DC    CL8'*'          -TAP1
        DC    CL8'TAP0'
        DC    CL8'*'          -TAP2
        DC    CL8'n'
        DC    CL8'*'          -1
        DC    CL8'yes'
        DC    CL8'*'          -no
```

OPERATION: TPCOPY calls SVCFREE to get free storage and then analyzes the parameter list looking for defaults and errors such as the same unit for both input and output, and a file number less than 1 and greater than 9.

It uses TAPEIO to read and write the tape records. If tape density is specified the corresponding modeset code is put in the TAPEIO parameter list for the output tape. TPCOPY calculates the actual length and number of records. This information is given to the user if the file summary option has been selected.

Error messages on read and write tape errors will be passed to the user by TPCOPY.

## WRTAPE

FUNCTION: To write a disk file onto tape.

ATTRIBUTES: Disk resident

CALLING SEQUENCE:

```
        LA    1, PLIST
        SVC   X'CA'
        .
        .
        .
PLIST   DC    CL8'WRTAPE'
        DC    CL8'        '     filename
        DC    CL8'        '     filetype
        DC    CL2'  '           filemode — defaults to P
        DC    CL8'        '     blocking factor — defaults to 10
        DC    CL3'EOF'          defaults to no-eof
```

# TAPE HANDLING FUNCTION PROGRAM

The following text describes the program that performs I/O operations to tape.

## TAPEIO

FUNCTION: To handle all CMS tape input/output operations

CALLING SEQUENCE:

```
    LA      1, PLIST
    SVC     202
    DC      AL4 (error routine)

PLIST  DC     CL8'TAPEIO'
       DC     CL8'function'       function code
       DC     CL4'TAPn'           tape number (n = 1 to 4)
       DC     XLI'mm'             modeset code
       DC     AL3(buffer)         I/O buffer address
       DC     F'buffer size'      buffer length (in bytes)
       DC     F'0'                number of bytes actually
                                  read returned here
```

where:

function is one of the following: READ, WRITE, REWIND, WRITEOF, WTM, RUN, BSR, FSR, BSF, FSF, or ERG.

mm is a modeset command code to be used in conjunction with a read, write, or write tape mark operation. If x'00' is specified, x'B3' (7-track, 800 bpi, odd parity, data converter off, translator off) is used for the mode. The default mode for 9-track tapes with the dual density feature installed is 1600 bpi.

EXIT CONDITIONS:

Normal Return

| | |
|---|---|
| R15 = 0 | Successful Operation |

Error Return

| | |
|---|---|
| R15 = 1 | Invalid Function |
| R15 = 2 | End-of-file or End-of-tape |
| R15 = 3 | Permanent I/O Error |
| R15 = 4 | Illegal Symbolic Unit |
| R15 = 5 | Tape is not attached |
| R15 = 6 | Tape is file protected |
| R15 = 7 | Serious tape error |
| R16 = 8 | Channel busy on start of I/O operation |

CALLED BY (WHERE KNOWN):

Disk resident routines

OPERATION: TAPEIO first checks the function code and tape number for validity. An error code of 1 indicates an invalid function and a code of 4 indicates an invalid tape number. TAPEIO then builds a CCW to do the requested function. If it is a read, write, or write tape mark operation, the CCW is chained from a modeset CCW. The modeset code in the parameter list is used as the command code, unless it is zero. If the modeset code is zero, x'B3' is used as the modeset command code. The I/O operation is then started. For all operations except for rewind and rewind and unload, if it is started successfully, WAIT is called to wait for the I/O interrupt. After a rewind or rewind and unload operation is started a control is immediately returned to the caller. After WAIT returns control, the CSW is checked for any error conditions. If no errors occurred, the number of bytes read or written is computed and stored back in the parameter list and return is made to the caller. If the channel is busy when the operation is started, a message is sent to the user and return is made to the calling program with an error code of 8. If the tape is not operational, a message is sent to the user, and the error code is set to 5 before returning to the caller. If a unit check occurs either when the I/O operation is started or when the interrupt is received, the tape unit is immediately sensed. If intervention required is indicated, a message is sent to the user and WAIT is called to wait for an I/O interrupt and then the operation is retried. If command reject is indicated a file protected message is sent to the user and the error code is set to 6. If any other error occurs on a read or a write operation the tape is backspaced one record and the operation is retried. If the I/O operation did not complete successfully after 10 retries, a message is sent to the user and the error code is set to 3. I/O errors on operations other than read or write and any errors encountered while sensing the device or backspacing during a retry cause the error code to be set to 7 and a message to be sent to the user.