ADDENDUM TO 9520 MANUAL
CONVERT UTILITY PROGRAM
PUBLICATION NO. 87000087
REV. JANUARY 1982

ADDENDUM TO 9520 MANUAL

CONVERT UTILITY PROGRAM

Convert Utility Program

To enhance the universality of the Millennium 9520 Software Development System, Millennium has installed a CONVERT utility program in the 9520 to run under the CP/M or MP/M operating system. The CONVERT utility enables the user to create a new file that will contain reformatted data from a specified file type. The following table denotes the available conversions.

Table 1

Conversions
T O

F R O M	BIN	HEX	TEK
BIN	*	X	X
HEX	X	*	X
OBJ	X	X	X
TEK	X	X	*

Legend: BIN = Millennium Binary Format Code
HEX = Intel Hexadecimal Format Code
TEK = Tektronix Hexadecimal Format Code
OBJ = Tektronix Object Code
X = Allowable Conversions
* = Not applicable

The CONVERT utility is invoked by entering the following instructions at the console terminal.

0A>CONVERT [d:]Fn1.Ft1 [d:]Fn2.Ft2 <cr>

Syntax

0A>CONVERT [drive:]output filename.file type extension
[drive:]input filename.file type extension

where:

[drive] = the disk drive A, B, C, or D. The default is the drive currently logged on.

output file name = the actual file name of the file that will contain the converted data.

file type extension = one of the conversion extensions (e.g., .BIN, .HEX, .OBJ, or .TEK).

input file name = the file name of the file containing the data to be converted.

space = delimiter

The Convert utility program flow and file name manipulation performs as follows:

a. Verify specified input and output file type extension.

b. Open specified input file ([d:]Fn2.Ft2), erase working output file ([d:]Fn1.$$$), and open working output file ([d:]Fn1.$$$).

c. Input data from specified input file ([d:]Fn2.Ft2), convert the data and output converted data to the working output file ([d:]Fn1.$$$).

d. Close specified input file ([d:]Fn2.Ft2) and close working output file ([d:]Fn1.$$$).

e. If errors or warnings occur during the conversion, then proceed to step 6; If not, erase specified output file ([d:]Fn1.Ft1), rename working output file ([d:]Fn1.$$$) to specified output file ([d:]Fn1.Ft1), output to the console "FILE CONVERTED, NO ERRORS", and return control to the operating system.

f. Output to the console "FILE CONVERTED, REMAINS ".$$$"," and return control to the operating system.

In the event conversion is not accomplished without errors, the CONVERT utility will cause (an) error message(s) to be displayed. There are two types of error conditions with associated error messages, warning error messages and fatal error messages.

Warning Error Messages. When the CONVERT utility encounters any impropiety in the input file that is not a fatal error, an error message is displayed and file conversion continues. The warning messages, as they are displayed on the console terminal, are as follows:

1. ERROR: CHECKSUM DOES NOT MATCH
   CHECKSUM IN FILE: XX, CHECKSUM COMPUTED: XX
   RECORD LOAD ADDRESS: XXXX

   (XX=Hexadecimal value) (XXXX=Hexadecimal address value)

2. ERROR: INPUT MODULE IS NOT ABSOLUTE, IT CONTAINS RELOCATION INFO
   RECORD LOAD ADDRESS: XXXX

3. ERROR: INPUT FILE HAS SHORT BLOCK
   RECORD LOAD ADDRESS: XXXX

4. ERROR: INPUT MODULE HAS UNRESOLVED EXTERNAL REFERENCE
   RECORD LOAD ADDRESS: XXXX

5. ERROR: UNDEFINED ERROR
RECORD LOAD ADDRESS: XXXX

6. ERROR: INTEL RECORD MARK (":") EXPECTED,
FILE CONTAINS: XXX--------XXX
XXX----------XXX
RECORD LOAD ADDRESS: XXXX

NOTE: For ERROR 6, XXX----------XXX (to 80 column width) represents the data read from the file until the next record mark was encountered. When converting Millennium Binary files, unprintable binary data is represented by a period (".") . In addition, spaces between data are also represented by a period (".").

7. NOTICE: INTEL RECORD TYPE 02 (EXTENDED ADDRESS RECORD) ENCOUNTERED
RECORD LOAD ADDRESS: XXXX

8. WARNING: INTEL RECORD TYPE 02 CONTAINS RECORD LENGTH OF XX
RECORD LOAD ADDRESS: XXXX

9. WARNING: EOF RECORD TYPE CONTAINED A NON-ZERO RECORD LENGTH OF XX
RECORD LOAD ADDRESS: XXXX

10. NOTICE: INTEL RECORD TYPE 03 (START ADDRESS RECORD) ENCOUNTERED
RECORD LOAD ADDRESS: XXXX

11. WARNING: INTEL RECORD TYPE 03 CONTAINS A RECORD LENGTH OF XX
RECORD LOAD ADDRESS: XXXX

12. ERROR: UNEVEN NUMBER OF BYTES TO CONVERT TO BINARY
RECORD LOAD ADDRESS: XXXX

13. ERROR: BAD RECORD TYPE (MRLLDADRT)
RECORD LOAD ADDRESS: XXXX

where - M=record mark (":" =INTELHEX record mark)
RL=record length
LDAD=load address
RT=record type

14. ERROR: TEK RECORD MARK ("/" or " ") EXPECTED
FILE CONTAINS: XXX------XXX
XXX----------XXX
RECORD LOAD ADDRESS: XXXX
(see ERROR 6 for definition of XXX----XXX for ERRORS 14,15,16,17.)

15. ERROR: BAD EXTEND-TEK RECORD TYPE: /XX
FILE CONTAINS: XXX------XXX
XXX----------XXX
RECORD LOAD ADDRESS: XXXX

16. ERROR: BAD EXTENDED-TEK SEGMENT ADDRESS: BC0000RTSSSS
FILE CONTAINS: XXX------XXX
XXX----------XXX
RECORD LOAD ADDRESS: XXXX

where - BC=byte count
0000=load address
RT=record type
SSSS=segment or offset address

17. ERROR: BAD EXTENDED-TEK OFFSET ADDRESS: BC0000RTSSSS
FILE CONTAINS: XXX------XXX
XXX----------XXX
RECORD LOAD ADDRESS: XXXX
(BC0000RTSSSS-same as ERROR 16)

18. NOTICE: EOF RECORD PROCESSED, BUT INPUT FILE STILL CONTAINS UNPROCESSED DATA

19. ERROR: EOF REACHED WITHOUT PROCESSING EOF RECORD

Fatal Error Messages. The following messages are output to the console display terminal when the CONVERT utility is aborted and program control is returned to the operating system.

1. ERROR: INVALID PARAMETER(S)

2. ERROR: INVALID INPUT FILE TYPE

3. ERROR: INVALID OUTPUT FILE TYPE

NOTE: ERROR 1, 2 and 3 is displayed when the .file-type-extension is not one of the previously mentioned file type extension(s), or when the file type extension(s) entered do not correspond to the types of conversions enumerated in the conversion table.

4. FATAL ERROR nn Fn.Ft RECORD=rr
OUTPUT ERROR

NOTE: The following definition is also valid for all remaining fatal errors.

where: nn = the error number
Fn = the file name
Ft = the file type. For input errors, it will be the file type of input file. For output errors, it will be the file type of the output file.
rr = the physical record number in the File Control Block (FCB).

5. FATAL ERROR nn Fn.Ft RECORD=rr
CANNOT OPEN FILE

6. FATAL ERROR nn Fn.Ft RECORD=rr
OUTPUT ERROR

7. FATAL ERROR nn Fn.Ft RECORD=rr
   INVALID BLOCK OR RECORD TYPE

8. FATAL ERROR nn Fn.Ft RECORD=rr
   FILE NOT OPEN

9. FATAL ERROR nn Fn.Ft RECORD=rr
   CANNOT CLOSE FILE

10. FATAL ERROR nn Fn.Ft RECORD=rr
    CANNOT RENAME FILE