WANG

3740 Diskette Compatibility Software Release 2 User Manual

3740 DISKETTE COMPATIBILITY SOFTWARE RELEASE 2 USER MANUAL

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851, TEL. (617) 459-5000, TWX 710 343-6769, TELEX 94-7421

Disclaimer of Warranties and Limitation of Liabilities

The staff of Wang Laboratories, Inc., has taken due care in preparing this manual; however, nothing contained herein modifies or alters in any way the standard terms and conditions of the Wang purchase, lease, or license agreement by which this software package was acquired, nor increases in any way Wang's liability to the customer. In no event shall Wang Laboratories, Inc., or its subsidiaries be liable for incidental or consequential damages in connection with or arising from the use of the software package, the accompanying manual, or any related materials.

NOTICE:

All Wang Program Products are licensed to customers in accordance with the terms and conditions of the Wang Laboratories, Inc. Standard Program Products License; no ownership of Wang Software is transferred and any use beyond the terms of the aforesaid License, without the written authorization of Wang Laboratories, Inc., is prohibited.

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

PREFACE

Wang's 3740 Diskette Compatibility Software simplifies programming requirements for applications where 3740 diskettes are to be accessed for data storage, retrieval, or updating by a Wang system -- if the system is equipped with a Model 2270A (IBM 3740 compatible) diskette drive. Also, the software may be used to convert program files from IBM 5110 diskettes to Wang 2200 diskettes.

The software resides on a diskette (#701-2212B) available in the 3740 Diskette Compatibility Software Package (#195-0041-3), which also contains a copy of this manual.

Chapters 1 and 2 in the manual present background information and software operating instructions of general interest. Chapter 3 presents information of primary interest to application programmers. For the information to be completely meaningful, readers of this manual should be familiar with the disk/diskette operations described in the Disk Reference Manual supplied with Wang systems having a 2200T, VP, or MVP central processor.

CONTENTS

FIGURES

TABLES

CHAPTER 1

GENERAL INFORMATION

1.1 SOFTWARE OVERVIEW

The 3740 Diskette Compatibility Software, designed for use with Wang's IBM 3740 Compatible Diskette Drive, provides a set of utility programs for:

- displaying 3740 diskette catalog information (each file's name, its reserved and used sector addresses, its record length, and its codes for write protect, bypass, and multivolume indicators);
- displaying 3740 sector dumps for a specified address range;
- converting 3740 diskette files to Wang disk/diskette files*, or vice versa;
- converting IBM 5110 program files to Wang disk/diskette files;
- producing hard copy of a specified 3740 diskette file or Wang diskette file previously converted from a 3740 diskette;
- copying one 3740 diskette to another 3740 diskette;
- initializing an unused, preformatted 3740 diskette for use on Wang 3740 Compatible Diskette Drives.

The software also includes a set of utility subroutines which handle the following operations:

- opening a new or existing file;
- reading, rereading, or writing a sector with or without code translation;
- skipping or backspacing to the last or first sector, to a specific address, or a specified number of sectors;
- writing or updating an "end of data" pointer position;

---

* The 3740 Diskette Compatibility Software processes TC formatted Wang files (maximum record length 128 bytes). See Appendix A.

- deleting the currently open file or the currently accessed record;
- setting a file's bypass, write protect, and multivolume indicator codes;
- closing a file.

Programmers may integrate these subroutines with their BASIC application programs which access 3740 diskettes for file creation or maintenance.

Wang Laboratories offers several types of disk and diskette drives in its product lines. The Model 2270A (IBM 3740 compatible) single, dual, or triple diskette units (unlike the Model 2270 units) have the necessary hardware and firmware to sense whether an IBM 3740 or a Wang diskette is currently mounted at a disk location where a read or write operation is to be executed; however, compatibility software is needed when data is to be stored on, retrieved from, or converted to a 3740 diskette by a Wang system.

1.2 MEDIA DIFFERENCES

IBM's 3740 Data Entry System encompasses such equipment as the single-operator 3741 Data Station and the dual-operator 3742 Dual Data Station. These "key to diskette" data stations store data on diskettes resembling the ones used in Wang systems. Each diskette is a thin, flexible disk platter 7.5 inches (19 cm) in diameter -- about the size of a 45 rpm phonograph record -- enclosed in an 8-inch by 8-inch (20 cm by 20 cm) semi-rigid protective plastic jacket. The diskette turns freely within the jacket and is coated on one side with magnetic material arranged in concentric circular tracks. When formatted, the tracks are divided into "sectors" with unique, randomly accessible addresses which provide a rapid, direct access method for data storage and retrieval. Although important differences usually preclude the exchange of diskette files between Wang and IBM systems, Wang's 3740 Compatible Diskette Drive and the software described in this manual allow 3740 data files and 5110 program files to be read into a Wang system for subsequent processing or use. Also, Wang data files can be stored on 3740 diskettes and used as input to any system equipped to read 3740 diskette files.

Visual Differences and Mounting Instructions

An elongated label with the name "Wang Diskette" and part number WLI No. 177-0063 identifies a diskette usable in Wang Model 2270 or 2270A type drives. The label has arrows marked "Insert" and "Up" showing how to position the diskette before mounting it in a drive (see Figure 1-1); a third arrow points to the "write protect" notch which prevents writing when uncovered, or permits writing when covered by a tab.

A square label with the name "IBM Diskette" and Part No. 2305830 identifies a diskette usable by 3741 and 3742 data station operators. No arrows show how to orient a 3740 diskette before mounting. To mount a 3740 or 5110 diskette in one of Wang's 3740 compatible drives, hold the diskette edgewise with the label on the right side, in the lower corner away from the drive door, as shown in Figure 1-2.

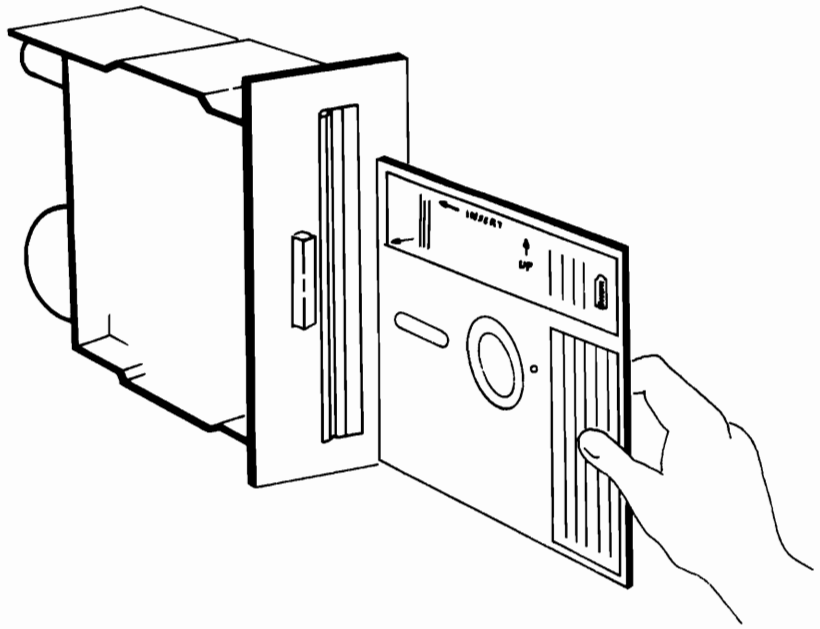


Figure 1-1. Mounting a Wang Diskette

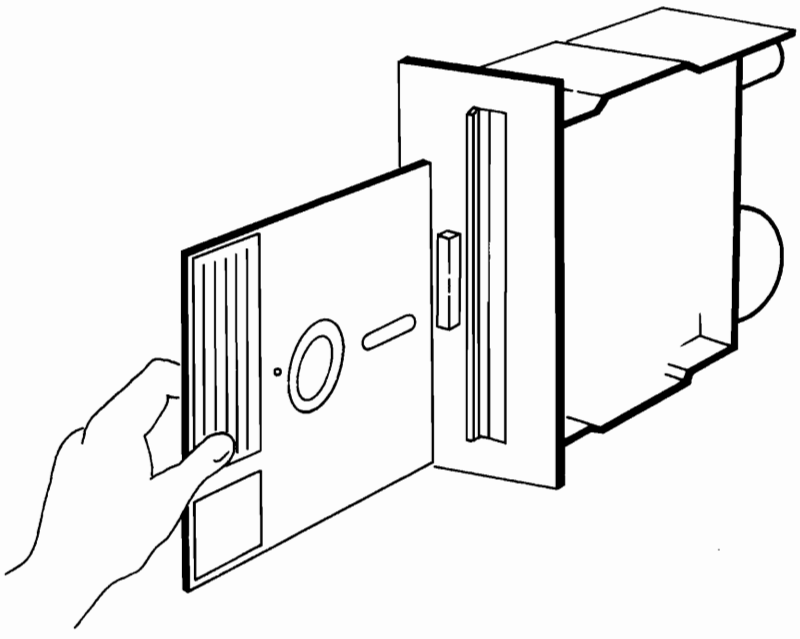


Figure 1-2. Mounting a 3740 Diskette

Other Significant Differences

For background information, other differences between 3740 and Wang diskettes are summarized here:

- A Wang diskette has 32 small index holes around its central mounting hole; a 3740 diskette has only one index hole.
- The tracks on Wang diskettes are divided into 16 equal sectors per track, with a storage capacity of 256 bytes per sector; the tracks on 3740 diskettes are divided into 26 equal sectors per track, with a storage capacity of 128 bytes per sector.
- Sector addressing notation differs. The first sector on Wang diskettes has address zero; the first sector on 3740 diskettes has address 16,384 in Wang notation. The 3740 sectors are arranged and numbered consecutively, but the sequentially numbered Wang sectors are staggered (interlaced) to increase performance during multisector read/write operations.
- Catalog index layouts differ; also, timing mark conventions (used to identify stored data files) differ.
- Data are recorded in different code sets -- Wang uses ASCII (American National Standard Code for Information Interchange) while IBM uses EBCDIC (Extended Binary Coded Decimal Interchange Code).
- Valuable data stored on a Wang diskette can be protected against accidental overwriting by uncovering the write protect notch; no manual technique for write protection exists on 3740 diskettes.
- A Wang diskette can be formatted in the leftmost drive by pressing the recessed format button; a 3740 diskette must be purchased preformatted or be formatted by a 3740 system if the diskette is to be used for data storage or retrieval in a Wang system.

Programming Differences

For Wang diskettes, read/write operations can be controlled via the disk statements described in the disk reference manual provided with a Wang system, or any available disk utility software can be used. All information pertaining to the Model 2270 single, dual, or triple diskette units can be applied unchanged to comparable units of Wang's 3740 Compatible Diskette Drive when using Wang diskettes to store or retrieve diskette files.

For 3740 diskettes, none of the Automatic File Cataloging Mode disk statements can control read/write operations, but six Absolute Sector Addressing Mode disk statements can accommodate the 3740 sector addresses which must be greater than or equal to 16,384; the statements are:

DATALOAD BA	DATALOAD DA	COPY
DATASAVE BA	DATASAVE DA	VERIFY.

Furthermore, since Wang's disk statements are designed to read and write information in 256-byte sectors rather than the 128-byte sectors formatted on 3740 diskettes, special programming techniques are needed to prevent loss of 128 bytes of valid data when information is transferred from the CPU to the disk controller for storage in a 3740 sector. Similarly, since the disk controller automatically supplies zero-fill information after reading a 128-byte 3740 sector, programming techniques must reflect the fact that valid data can lie only in the first 128 bytes of a Wang sector resulting from the reading of a 3740 sector. Also, since the disk controller supplies a nonzero value in the 129th byte if a deleted or bad 3740 sector is read, a check of the 129th byte of a sector should be performed before processing the first 128 bytes of information. During a writing operation to a 3740 diskette, the 129th byte should be set to HEX(00) as a preferred technique.

Because the differences between Wang and 3740 diskettes impose programming constraints of considerable magnitude, Wang Laboratories, Inc., recommends use of its 3740 Compatibility Software to simplify application programming requirements associated with data storage or retrieval via 3740 diskettes. With the compatibility software, differences between 3740 and Wang diskettes become transparent to the programmer.

1.3 DISKETTE STORAGE CONSIDERATIONS

On 3740 diskettes, there are 77 tracks numbered from 0 through 76, beginning with the outermost track; each track is divided into 26 sectors with a storage capacity of 128 bytes per sector. Track 0, called the index track, is reserved for information describing the diskette's contents; the last two tracks (75 and 76) are reserved for use as replacements for defective tracks. Thus, the maximum space utilized for storage of data files (called data sets by IBM) is 74 by 26 by 128 or 246,272 bytes.

The "extent" of each named data set is given in the 3740 index track in the form of three addresses. See Figure 1-3 and Appendix B.

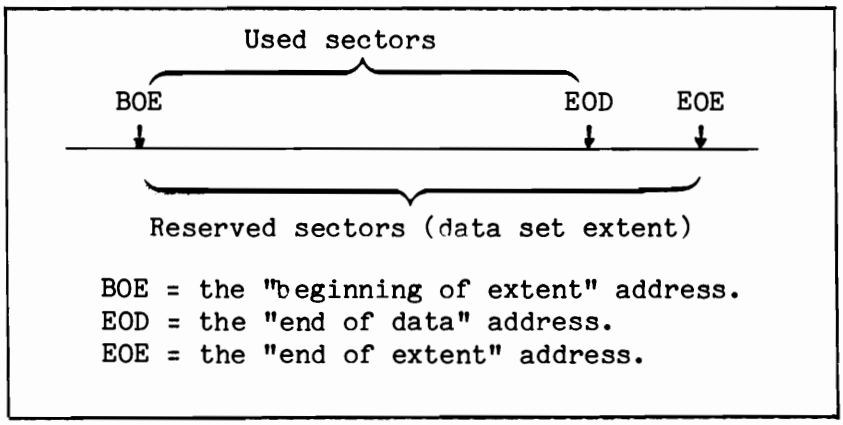


Figure 1-3. Storage Addresses for 3740 Files

On Wang diskettes, there are 77 tracks numbered from 0 through 76, and each track is divided into 16 sectors with a storage capacity of 256 bytes per sector. Only 64 tracks are accessible for disk operations via Model 2270 type diskette units, but all 77 tracks are accessible via Model 2270A type diskette units if the SCRATCH DISK statement parameter END is set to 1231 instead of 1023 when initializing a Wang diskette.

Also, the reserved catalog index space on a Wang diskette is selectable via the value assigned to the parameter LS in the SCRATCH DISK statement used to initialize the diskette. Since any number of sectors from 1 to 255 (24 is the default value) can be reserved, the maximum number of bytes available for storage of files on Wang diskettes varies as follows:

A = 262144 - 256*n (for Model 2270 units)

= 315392 - 256*n (for Model 2270A units)

where n = the value assigned to LS when scratching a diskette.

For media conversion from a 3740 diskette to a Wang diskette, only two sectors are needed for the Wang catalog index. The 3740 index track can hold a maximum of 19 file entries. The first sector in the Wang diskette catalog index can hold 15 file entries; any other sector can hold 16 file entries.

> NOTE:
>
> All diskettes formatted in Model 2270 diskette units are "upward" compatible with Model 2270A units. If "downward" compatibility between Model 2270A and 2270 units is desired, do not take advantage of the 77 track read/write feature when scratching (initializing) a Wang diskette in a Model 2270A drive, i.e., set END = 1023, not 1231.

1.4 MIXED MEDIA ERROR CONDITIONS

The disk controller in Wang's 3740 Compatible Diskette Drive senses which type of diskette (Wang or 3740) is mounted in a drive by the number of index holes on the diskette, and recognizes whether a disk read or write operation applies to a Wang or a 3740 diskette by the sector address (i.e., whether the address is less than 16,384 or not). Accordingly, an error code (ERR 64) interrupts a disk operation if one of the following conditions occurs:

a) a Wang diskette operation (sector address less than 16,384) is attempted on a 3740 diskette (only one index hole), or

b) a 3740 diskette operation (sector address equal to or greater than 16,384) is attempted on a Wang diskette (32 index holes).

Then, corrective action can be taken.

1.5 CPU AND PERIPHERAL REQUIREMENTS

The minimum number of disk drives required when using Wang's 3740 Diskette Compatibility Software is two. One Model 2270A (IBM 3740 compatible) drive is needed for mounting the 3740 diskette to be used for data storage or retrieval. Another 2270A drive, or any other disk/diskette drive in the system, is needed for mounting a platter containing the utility software or a user-written program. If a system has a 3740 compatible single diskette drive, the unit must be used in conjunction with a fixed/removable disk drive, or must be upgraded to a dual diskette drive unit.

The central processor of the Wang system must be a 2200T, VP, MVP, or the equivalent to one of these units. Approximately 10K bytes of memory are needed to operate the largest utility program in the software system. The subroutines, if appended to a user-written application program, require 3016 bytes of memory when loaded or 3874 bytes after a RUN command is executed. The difference between the "load only" and the "load run" requirements represents the space for variables and may interest a 2200MVP programmer whose application program must be optimized with respect to partitioned memory requirements.

A printer is needed when: (1) a hard copy of a 3740 diskette file or a Wang file (converted from a 3740 diskette) is desired, (2) the "all files" mode is chosen during execution of a media conversion utility, or (3) the hard copy option is chosen during execution of the sector dump utility.

1.6 SOFTWARE BACKUP

Before attempting to use the 3740 Compatibility Software for the first time, copy the software onto another diskette and carefully label the duplicate platter. Using the duplicate platter ensures that the master platter is always available as a backup system.

The COPY statement, described in the disk reference manual, may be used to copy the contents of one platter onto another (if both platters are mounted in a dual drive unit). Also, Wang's Integrated Support System (ISS) includes a utility which can be used to copy the 3740 Compatibility Software to a disk cartridge, if desired.

CHAPTER 2

OPERATING INSTRUCTIONS

2.1 LOADING THE SOFTWARE SYSTEM

Mount the 3740 software disk/diskette; then, enter the following sequence:

```
CLEAR
SELECT DISK xyy          (replace xyy by the proper address)
LOAD DC T "START"
RUN
```

or use the convenient LOAD RUN command available in systems with Wang's BASIC-2 language.

Displays indicating that the loading operation is in progress appear first. No operator action is required until a prompt requests the addresses to be used for: (1) program loading, (2) reading/writing Wang TC formatted files, and (3) reading/writing 3740 (3741) files. If specifying addresses other than the default addresses, remember that the program loading and TC formatted files' addresses may be the same; the 3740 files' address must be unique. After the operator either accepts the default addresses or specifies different addresses, the system menu appears on the CRT. (See Figure 2-1.)

```
PRESS DESIRED FUNCTION KEY _
3741-2200 UTILITY SYSTEM

-------------------------------------------------------------
FN KEY    DESCRIPTION            FN KEY    DESCRIPTION
-------------------------------------------------------------
   01 - CONVERT 3741 TO TC FORMAT  06 - LIST 3741 CATALOG
   02 - CONVERT TC FORMAT TO 3741  07 - DUMP 3741 SECTOR(S)
   03 - LIST 3741 FILE             08 - APPLICATION/SUBROUTINES
   04 - LIST TC FORMAT FILE        09 - CHANGE DISK ADDRESSES
   05 - CONVERT 5110 TO TC FORMAT  10 - INITIALIZE 3741 DISK
                                   11 - COPY 3741 DISK
   15 - RESTART UTILITY            31 - RETURN TO 'START'
```

Figure 2-1. The System Menu

2.2 THE UTILITY PROGRAMS

The system menu displays the available utilities and the Special Function (SF) keys by which they are accessed. After an operator accesses a particular program, instructions (prompts) appear on the CRT to indicate when disk platters are to be mounted and when input is required. No operational difficulties should arise if the prompts are followed carefully. For convenience, a summary of each program and its required input is given here.

> NOTE:
>
> If an unused, preformatted 3740 diskette is to be accessed by a Wang system, execute the "initialize 3741 disk" utility program first. The utility is not needed for a diskette which already contains data files.

CONVERT 3741 TO TC FORMAT (SF'1) -- prompts the operator to specify the desired choices from the available options, and then:

1. converts a single 3740 (3741) file or, optionally, all the files on the input platter to Wang TC format. Note: The maximum record length is 128 bytes.
2. automatically assigns a unique name to each output file if the ALL mode is in effect. Note: Unlike Wang file names, 3740 file names may exceed eight characters. If a 3740 file name is less than or equal to eight characters, that name becomes the initially assigned name for the corresponding output file; if the name exceeds eight characters, only the first eight characters become the initially assigned output file name. In either case, the output platter is checked to determine the uniqueness of the initial choice. If the name already exists on the output platter, another name is formed by retaining the first six characters of the input file name and appending the number 01 as the seventh and eighth bytes; then, the output platter is checked again. If necessary, the seventh and eighth bytes are changed to 02 and another check is made. This process continues until a unique name is generated.
3. accommodates any number of records in a file by implementing a multivolume feature when needed. Note: If there are at least three available sectors on the output platter, but insufficient room to allocate an entire file, a prompt shows the number of available sectors and number of required sectors. After the operator indicates whether to use the available sectors or a newly mounted platter, the system converts as many output records as possible. If the platter becomes full, the system closes the current volume of the file and displays a message requesting another output platter. The process continues until the entire input file is converted.
4. creates fixed-length records by adding space characters, if necessary, to satisfy the record length specified in the 3740 index track.

5. during file conversion, translates the records to ASCII or optionally leaves them in EBCDIC.
6. during file conversion, indicates the record currently being processed.
7. stops the conversion at EOD (End Of Data) or, optionally, at EOE (End Of Extent).
8. for the ALL mode, produces hard copy of the input file names and the corresponding output file names, as well as the names of any files not converted because of a lack of data.
9. provides an option to rerun the program before returning to the system menu.

Required information: (1) the mode (ALL or SINGLE FILE), (2) the translation option (translate to ASCII or leave in EBCDIC), (3) where to stop the conversion (EOD or EOE), and (4) for the SINGLE FILE mode only, the input and output file names.

CONVERT TC FORMAT TO 3741 (SF'2) -- prompts the operator to specify the desired choices from the available options, and then:

1. converts a single Wang TC formatted file or, optionally, all the files on the input platter to the 3740 (3741) format. Note: Records longer than 128 bytes are truncated.
2. automatically assigns unique names to the output files if the ALL mode is in effect. Note: Initially, the program assigns the eight-byte input file name to the output file. If this name already exists on the output platter, a ten-byte name is generated by appending the number 01 to the eight-byte name; then, the output platter is checked again. If necessary, the number is changed to 02 and another check is made. This process continues until a unique name is generated.
3. accommodates any number of records in a file by implementing a multivolume feature when needed. Note: If there are less than two available sectors on the output platter, a message appears and the system waits for the operator to mount another platter. Once an output platter with at least two available sectors is mounted, the system converts as many output records as possible. If the platter becomes full, the system closes the current volume of the file, sets the multivolume label code to "C" for the file's entry in the platter's index tract, and displays a message requesting another output platter. This process continues until the entire input file is converted and the file's multivolume label code is set to "L" to denote the last volume.
4. during file conversion, translates the records to EBCDIC or optionally leaves them in ASCII.
5. during file conversion, indicates the record currently being processed.
6. for the ALL mode, produces hard copy of the input file names and the corresponding output file names.
7. provides an option to rerun the program before returning to the system menu.

Required information: (1) the mode (ALL or SINGLE FILE), (2) the translation option (translate to EBCDIC or leave in ASCII), and (3) for the SINGLE FILE mode only, the input and output file names.

LIST 3741 FILE (SF'3) -- provides hard copy of the 3741 file on a printer with address 215. Records are read in EBCDIC or, optionally, in ASCII and dumped with no attempt to format the data; nonprintable characters are printed as question marks (?). The 3741 file is listed from BOE to EOE, i.e., from the beginning to the end of the reserved sectors. An end-of-data record, if encountered, is also listed and looks like the index entry with the name blanked. Required information: (1) the 3741 input file name, (2) where to stop the listing (EOD or EOE), and (3) the read option, (read EBCDIC or ASCII).

LIST TC FORMAT FILE (SF'4) -- provides hard copy of a Wang TC formatted file on a printer with address 215. Records are read in ASCII or, optionally, in EBCDIC and dumped with no attempt to format the data; nonprintable characters are printed as question marks (?). Required information: (1) the Wang input file name and (2) the read option (read ASCII or EBCDIC).

CONVERT 5110 TO TC FORMAT (SF'5) -- prompts the operator to specify the desired choices from the available options, and then:

1. converts a single 5110 program file or, optionally, all the 5110 files on the input platter to Wang TC format.
2. automatically assigns unique names to the output files if the ALL mode is in effect. Note: Unlike Wang file names, 5110 file names may exceed eight characters. If a 5110 file name is less than or equal to eight characters, that name becomes the initially assigned name for the corresponding output file; if the name exceeds eight characters, only the first eight characters become the initially assigned output file name. In either case, the output platter is checked to determine the uniqueness of the initial choice. If the name already exists on the output platter, another name is formed by retaining the first six characters of the input file name and appending the number 01 as the seventh and eighth bytes; then, the output platter is checked again. If necessary, the seventh and eighth bytes are changed to 02 and another check is made. This process continues until a unique name is generated.
3. accommodates any number of records in a file by implementing a multivolume feature when needed. Note: If there are less than three available sectors on the output platter, a message instructs the operator to mount another output platter. If the output platter contains at least three available sectors, the system opens the output file to the extent of the catalog area. The conversion continues until either the end of the input file is reached or no room remains on the output platter. If the latter case occurs, a message requests mounting of another output platter. After the entire input file is converted, the system closes the current volume of the file and frees any unused sectors.
4. creates fixed-length records by adding space characters, if necessary, to satisfy the record length specified in the 5110 index track.
5. during file conversion, translates the records to ASCII or optionally leaves them in EBCDIC.

6. during file conversion, indicates the record currently being processed.
7. stops the conversion only at EOD since any data encountered beyond the end trailer is irrelevant and not recognizable by the TC Support Utilities.
8. for the ALL mode, produces hard copy of the input file names and the system-assigned output file names, as well as the names of any files not converted because they were not 5110 program or data files.
9. provides an option to rerun the program before returning to the system menu.

Required information: (1) the mode (ALL or SINGLE FILE), (2) the translation option (translate to ASCII or leave in EBCDIC), and (3) for the SINGLE FILE mode only, the input and output file names.

LIST 3741 CATALOG (Index Sectors) (SF'6)--displays a list of the files on the currently mounted 3741 diskette. In addition to the names of the files, numbers under the heading LEN indicate the record length in bytes, and codes under the heading BPV represent the bypass, write protect, and multivolume indicators described in Appendix B. Also, numbers under the headings 'BOE', 'EOE', and 'EOD' indicate the sector addresses corresponding to the beginning of extent (the first sector in the file), the end of extent (the last sector reserved for the file), and the end of data (the next unused sector within the file); numbers in parentheses under each heading indicate the 3741 track and the sector position within the track. Note: The numbers given without parentheses are appropriate for use with the software system when dumping sectors in a particular file (see Figure 2-2). Required information: none.

NAME	LEN	'BOE'	'EOE'	'EOD'	BPV
SAMPLE	80	26 (01/01)	113 (04/10)	35 (01/10)	B
FILE010A	128	114 (04/11)	243 (09/10)	237 (09/04)	PC
FILE030A	100	244 (09/11)	1056 (40/17)	1023 (39/10)	L
################	###	#### (##/##)	##### (##/##)	##### (##/##)	###

(Note: The print format, shown in the last line, does not appear in an actual display.)

Figure 2-2. A Sample 3740 Catalog Listing

DUMP 3741 SECTOR(S) (SF'7) -- for a specified range of sector addresses, reads and displays the 3741 sectors with no attempt to format the data and with nonprintable characters represented by question marks (?). The sectors are read in EBCDIC or, optionally, in ASCII. Each sector's contents are displayed in a triple line format. The first line shows the actual characters, the second line shows the high-order hexadecimal digit corresponding to each character, and the third line shows the low-order hexadecimal digit for each character. (For example, if the ASCII-coded character B is in a sector's fifth byte and the sector is read in ASCII, a B appears in the fifth position of the first line of

the displayed sector; beneath the B, the digit 4 appears in the second line and the digit 2 appears in the third line since the ASCII code for B is 42 in hexadecimal notation.) The sector dump remains on the screen until the operator touches the SF key which initiates one of the following actions:

- SF'4 -- dumps the last sector of the specified range.
- SF'5 -- changes the read option to ASCII if currently EBCDIC, or to EBCDIC if currently ASCII.
- SF'7 -- dumps the first sector of the specified range.
- SF'8 -- dumps a particular sector of the specified range (after the operator supplies the sector address in response to "ENTER SECTOR TO DUMP").
- SF'9 -- redisplays the current sector.
- SF'10 -- produces hard copy of the current sector. (If the printer is not selected, a message alerts the operator while the system waits until the printer is selected.)
- SF'11 -- skips to the fifth sector beyond the current address (or to the last sector in the specified range if the new address is outside the range) and dumps that sector.
- SF'12 -- skips to the next sector (unless the current sector is the last sector in the range) and dumps that sector.
- SF'13 -- backspaces to the preceding sector (unless the current sector is the first sector in the range) and dumps that sector.
- SF'14 -- backspaces to the fifth sector preceding the current address (or to the first sector in the specified range if the new address is outside the range) and dumps that sector.
- SF'15 -- restarts the utility.
- SF'31 -- recalls the system menu.

Required information: (1) the read option (read in ASCII or EBCDIC) and (2) the sector range (any pair of numbers between 1 and 1950, where a suitable range of sector addresses for a particular file may be determined from the LIST 3741 CATALOG utility).

APPLICATION/SUBROUTINES (SF'8) -- accesses a sample application program if the operator uses file name 3741090A as the response to the system's request for the name of the application program; then, via a new menu (see Figure 3-1) and prompts, the sample program demonstrates the file maintenance capabilities of the utility subroutines. Note: These subroutines simulate several Wang disk operations while compensating for the differences between Wang and 3740 diskettes. Use of the sample application program to maintain files is not recommended; see Chapter 3 for a description of each subroutine and the requirements for integrating the subroutines with user-written application programs. Required information: the name of an application program stored on the platter with the 3740 software.

CHANGE DISK ADDRESSES (SF'9) -- allows the operator to change the default disk addresses. If three disk drives are available, the specified address for program loading should differ from the address for reading/writing Wang files (to eliminate the necessity of removing the software system platter each time a Wang platter is to be mounted). Required information: any new addresses to replace the current default values.

INITIALIZE 3741 DISK (SF'10) -- prepares a preformatted, unused 3740/3741 diskette for use on Wang 3740 Compatible Diskette Drives (by deleting sector 8 in the index track); otherwise, files begin on track 74 instead of track 1. Note: Do not initialize a 3740 diskette which already contains data files. Required information: none. (When prompted to mount the diskette, use the drive corresponding to the specified address for reading/writing 3741 files.)

COPY 3741 DISK (SF'11) -- allows the operator to copy a set of sectors from one 3741 diskette to another 3741 diskette, beginning with sector 8 and extending through a specified sector. The copy is verified automatically. Note 1: Previously specified mounting addresses do not apply to this utility. By selecting FR or RF, the operator indicates whether the copying operation should occur from the F to the R platter or vice versa. Note 2: All new files are opened at the greatest EOE of all active files on the diskette receiving the copied sectors. The utility does not support diskette reorganization. To reorganize a diskette requires three phases: (1) convert all desired 3741 files to TC format, (2) delete all active files on the 3741 diskette, and (3) convert the TC formatted files back to the 3741 format. Required information: the address of the last sector to be copied.

RESTART UTILITY (SF'15) -- provides a convenient method of restarting the currently executing utility program if an inappropriate value is entered in response to a prompt. Exception: During execution of the sample application program (3741090A), SF'15 recalls the application menu.

RETURN TO 'START' (SF'31) -- recalls the system menu and, thereby, provides access to a different utility. Exception: During execution of the sample application program, SF'31 reloads the software system.

CHAPTER 3

THE UTILITY SUBROUTINES

3.1 GENERAL DESCRIPTION

As indicated in Sections 1.1 and 2.2, a set of utility subroutines in Wang's 3740 Diskette Compatibility Software can be integrated with user-written BASIC or BASIC-2 application programs where 3740 diskettes are to be accessed directly for file maintenance purposes. These subroutines simulate several of Wang's BASIC language disk operations as indicated in the following alphabetized list of subroutine names:

Subroutine Name	Equivalent Operation
BACKSPACE	DBACKSPACE
CLOSE	DATASAVE DC CLOSE
DELETE FILE	SCRATCH name
DELETE RECORD	none
OPEN (NEW)	DATASAVE DC OPEN
OPEN (OLD)	DATALOAD DC OPEN
READ WITH TRANSLATION	DATALOAD DC
READ WITHOUT TRANSLATION	DATALOAD DC
REREAD	DBACKSPACE 1 plus DATALOAD DC
SET LABEL CODES	none
SKIP	DSKIP
SKIP TO ADDRESS	none
SKIP TO EOD	DSKIP END plus DBACKSPACE 1
WRITE WITH TRANSLATION	DATASAVE DC
WRITE WITHOUT TRANSLATION	DATASAVE DC
WRITE END	DATASAVE DC END

At the system menu stage, shown in Figure 2-1, SF'8 provides the first step in the linkage between the utility subroutines and an application program residing on the platter with the software. If SF'8 is depressed, a prompt appears on the CRT requesting the name of an application program. No further action occurs until the operator supplies a program name, or depresses SF'31 to return to the system menu.

A sample application program (file 3741090A), described in Section 3.2, is included in Wang's compatibility software. The program should be viewed as an example of how to integrate an application program with the software system; use of the program to maintain 3740 files is not recommended.

The requirements for integrating a user-written application program are given in Section 3.3. Anyone writing an application program has two alternatives:

1. The program can be stored on a platter with the software system (see Section 1.6).

2. The subroutines, i.e., file 3741000A, can be loaded into memory from the software platter, and then saved on a platter containing an application program.

The utility subroutines are described in alphabetical order. Each description includes steps indicating the priorities for error codes which may occur during execution of the subroutine. The error codes are listed in Table 3-3; the subroutines and their identifying numbers are listed in Table 3-1.

BACKSPACE -- positions the record pointer a specified number of sectors prior to its current position in an open 3740 file (or to the beginning of the file if the number is too large). Executes the following steps:
1. Change the sign of the specified number of sectors.
2. Go to the SKIP subroutine.

CLOSE -- clears the file open flag, thereby inhibiting access to all the subroutines except LIMITS, OPEN (OLD), and OPEN (NEW).

DELETE FILE -- changes the label ID control byte from HEX(00) to HEX(F8) and stores "D" in the first byte of the index track sector corresponding to the currently open file. Executes the following steps:
1. Check for file open.
2. Set file label control byte to "D".
3. Close file.

DELETE RECORD -- changes the control byte in the currently flagged record (sector) of the currently open file from HEX(00) to HEX(F8) and sets the first byte to "D". Executes the following steps:
1. Check for file open.
2. Set the first byte of the currently flagged record to "D".

LIMITS -- finds the BOE, EOE, and EOD sector addresses of a 3740 file, and indicates whether a file exists. Executes the following steps:
1. Set flags and pointers to give the appearance that a file has been opened with BOE = 8 and EOE = 26.
2. Read catalog sector.
3. Set catalog sector address pointer.
4. Check if unused sector.
5. Find the EOE address for this entry.
6. Check for match on file name.
7. Find the BOE and EOD addresses, the record length, and label codes.

OPEN (NEW) -- opens a new 3740 file. Executes the following steps:

1. Check record length.
2. Call LIMITS subroutine.
3. Check for file already cataloged.
4. Check for index full.
5. Check for disk full.
6. Save addresses of BOE, EOD, and EOE.
7. Set current record pointer to BOE.
8. Save file name and record length.
9. Set file open flag.

OPEN (OLD) -- opens an existing 3740 file. Executes the following steps:

1. Call LIMITS subroutine.
2. Check for existence of file.
3. Set current record pointer to BOE.
4. Set file open flag.

READ WITH TRANSLATION -- reads a record in EBCDIC from an open 3740 file and translates it to ASCII. Executes the following steps:

1. Check for file open.
2. Check for end of file.
3. Read next sector, assuming EBCDIC coded information.
4. Translate to ASCII code.
5. Check for deleted or bad sector; if either is true, go to step 2.
6. Transfer to record buffer, Z$().

READ WITHOUT TRANSLATION -- reads a record in ASCII from an open 3740 file. Executes the following steps:

1. Check for file open.
2. Check for end of file.
3. Read next sector, assuming ASCII coded information.
4. Check for deleted or bad sector; if either is true, go to step 2.
5. Transfer to record buffer, Z$().

REREAD -- reads the record prior to the current record pointer position. This routine may be used to perform a read after write. Executes the following steps:

1. Decrease the current record pointer by 1.
2. Go to READ WITH TRANSLATION routine.

SET LABEL CODES -- sets specified codes for bypass, write protect, and multivolume indicators in the index track sector corresponding to the currently open file. Executes the following steps:

1. Check for file open.
2. Set codes as specified, where valid codes for bypass are B or blank (" "), for write protect are P or blank, and for multivolume indicators are C, L, or blank, as described in Appendix B.

SKIP -- positions the record pointer a specified number of sectors beyond its current location in an open 3740 file (or to the end of the file if the number is too large). Executes the following steps:

1. Check for file open.
2. Check for positioning beyond the file's limits; if either the beginning or end would be exceeded, adjust the increment to an appropriate value.
3. Update current record pointer.

SKIP TO ADDRESS -- positions the record pointer to the specified record (sector) if its address is within the range of the currently open file; otherwise, an error code is returned. Executes the following steps:

1. Check for file open.
2. Check the range compatibility.
3. Set current record pointer to specified address.

SKIP TO EOD -- positions the record pointer to the last sector of active data (EOD minus 1). If the EOD has never been updated and is still equal to BOE, an error code is returned. Executes the following steps:

1. Check for file open.
2. Check for EOD equal to BOE.
3. Set current record pointer to EOD minus 1.

WRITE WITH TRANSLATION -- writes a record in EBCDIC to an open 3740 file. Executes the following steps:

1. Check for file open.
2. Check for file full.
3. Transfer record buffer to sector buffer.
4. Translate sector to EBCDIC.
5. Write sector.
6. Increase current record pointer.

WRITE WITHOUT TRANSLATION -- writes a record in ASCII to an open 3740 file. Executes the following steps:

1. Check for file open.
2. Check for file full.
3. Transfer record buffer to sector buffer.
4. Write sector.
5. Increase current record pointer.

WRITE END -- updates the open file's EOD indicator in the 3740 catalog. Executes the following steps:

1. Check for file open.
2. Check for file full.
3. Set EOD pointer to current record position.
4. Read catalog sector.
5. Update EOD bytes in catalog sector.
6. Write catalog sector.
7. Set current record pointer to EOD.

3.2 A SAMPLE APPLICATION PROGRAM

To initiate the sample application program, select the APPLICATION/SUBROUTINES option from the system menu, and mount the platter containing the program at the program loading address specified in the software system. Then, in response to a system prompt requesting the name of the application program, enter the name 3741090A.

The sample application program provides a menu of the available subroutines, as they existed in the previous release of the software system. (See Figure 3-1.) The READ routine in the menu is the routine named READ WITH TRANSLATION in this release. Similarly, the WRITE routine in the menu is the WRITE WITH TRANSLATION routine in this release. The sample application program does not provide access to the either the READ WITHOUT TRANSLATION or the WRITE WITHOUT TRANSLATION routine.

After a subroutine is accessed via its corresponding SF key, prompts indicate the information needed for the parameters associated with the subroutine. See Table 3-1 for a list of the subroutines and their parameters. This table includes the READ WITHOUT TRANSLATION and the WRITE WITHOUT TRANSLATION subroutines.

Depressing SF'15 while entering parameters causes the sample application menu to be redisplayed. Any errors returned during execution of a subroutine are printed in hexadecimal notation with the meaning of the error code beside the HEX value. (See Table 3-3.)

```
PRESS DESIRED FUNCTION KEY _

                              3741-2200 UTILITY SYSTEM

-------------------------------------------------------------
 FN KEY    DESCRIPTION          FN KEY    DESCRIPTION
-------------------------------------------------------------
   01 - READ                      08 - WRITE END
   02 - WRITE                     09 - OPEN (OLD)
   03 - BACKSPACE                 10 - OPEN (NEW)
   04 - SKIP                      11 - CLOSE
   05 - SKIP TO EOD               12 - DELETE RECORD
   06 - SKIP TO ADDRESS           13 - DELETE FILE
   07 - REREAD                    14 - SET LABEL CODES
   15 - RETURN TO MENU            31 - LOAD 'START'
```

Figure 3-1. The Sample Application Program Menu

3.3 REQUIREMENTS FOR INTEGRATING AN APPLICATION PROGRAM

A user-written application program, designed to access and/or maintain 3740 files, can be stored on a platter containing the 3740 Diskette Compatibility Software if the following requirements are satisfied by the program:

1. DEFFN' subroutine numbers listed in Table 3-1 must not be used to identify DEFFN' subroutines defined in the application program.

2. Variables listed in Table 3-2 must be treated as read only by the application program.

3. The application program must start at line 3000; however, if COM variables must be allocated, lines 11 through 19 may be used.

4. The following device numbers, set by the software system, must not be changed by the application program:

   #1, the program loading address,
   #5, the TC formatted files address, and
   #6, the 3741 files address.

   (Note: The subroutines use #6 for all disk access operations.)

5. To complete the linkage between a user-written application program and the utility subroutines, the application program should provide a means of loading a module named START when an operator keys SF'31, or when execution of the application program ends.

Once the user-written application program is stored on a platter with the 3740 software, the program can be initiated by selecting the APPLICATIONS/ SUBROUTINES option from the system menu, and supplying the appropriate name in response to the prompt requesting the name of the application program.

If a need arises to include the 3740 utility subroutines in an application program (rather than storing the application program on a platter containing the 3740 software), LOAD the file named 3741000A into memory from the platter containing the software system; then, SAVE the file on the platter containing the application program.

> NOTE:
>
> An application program should ensure that all sectors between a 3741 file's BOE and EOD addresses have been written on; otherwise, when IBM equipment attempts to read the file, a length error occurs.

Table 3-1. Subroutine Identification Numbers and Variables

Subroutine Name	Identification and Parameters	Type of Information to be Passed
REREAD	DEFFN' 180	None
READ W/TRANS	DEFFN' 181	None
WRITE W/TRANS	DEFFN' 182	None
BACKSPACE	DEFFN' 183 (N)	Number of sectors to backspace
SKIP	DEFFN' 184 (N)	Number of sectors to skip
WRITE END	DEFFN' 185	None
OPEN (OLD)	DEFFN' 186 (F$)	Name of file to open
OPEN (NEW)	DEFFN' 187 (F$,N,L)	Name of file to create, number of sectors to allocate, and record length
CLOSE	DEFFN' 188	None
LIMITS	DEFFN' 189 (F$)	File name
SKIP TO EOD	DEFFN' 190	None
SKIP TO ADDRESS	DEFFN' 191 (A)	Desired address
DELETE RECORD	DEFFN' 192	None
DELETE FILE	DEFFN' 193	None
SET LABEL CODES	DEFFN' 194 (C1$,C2$,C3$)	Bypass, write protect, and multivolume codes
READ WO/TRANS	DEFFN' 197	None
WRITE WO/TRANS	DEFFN' 198	None

Table 3-2. Subroutine Variables

Name and Dimension	Description
Z	Work
Z0	Current position + 1
Z1	BOE of file
Z2	EOE of file
Z3	EOD of file
Z4	Catalog sector for file
Z5	Record length
Z6	File open flag (0 = closed; 9 = open)
Z7	Number of sectors in file
Z8	Work
Z$17	File name
Z1$1	Bypass code
Z2$1	Write protect code
Z3$1	Multivolume code
Q$1	Error return code (See Table 3-3.)
Z$(2)64	Record buffer
Z0$(4)64	Sector buffer
Z9$(8)32	Translation table

Table 3-3. Subroutine Error Codes

Error Code, in Q$*	Description
00	Catalog end error
01	End file encountered
02	Record length greater than 128
03	File full
04	File already cataloged
05	File not found
06	Index full
07	No file open
08	Deleted record
09	Byte 129 equal to HEX(F8) and byte 1 not equal to "D" (Normally indicates deleted sector.)

* All error codes are expressed as hexadecimal values.

APPENDIX A

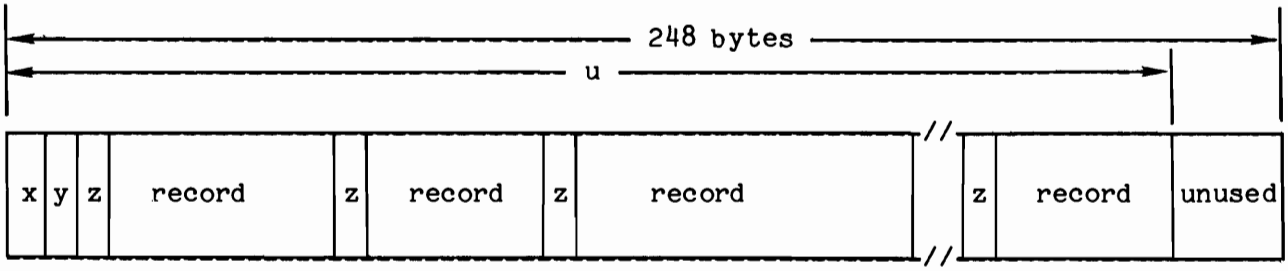
THE TC FILE FORMAT

The "TC file format" for disk data files is defined as follows:

1. Records are packed into a one-dimensional alphanumeric array with four elements, each 62 bytes long, e.g., DIM A$(4)62. Element boundaries are ignored; there are 248 contiguous bytes of storage.

2. The array is saved into a single sector by using a DATASAVE DC or DATASAVE DA statement.

3. Within the 248 bytes of storage, three types of control bytes are used (see x, y, and z in the following diagram).



x = a one-byte hexadecimal code indicating whether the sector is or is not the last sector in the file; in particular, x = HEX(F0) denotes "is the last sector", x = HEX(00) denotes "not the last sector".

y = a one-byte hexadecimal value denoting "the number of used bytes plus one" in the array (e.g., y = u + 1 for the above diagram).

z = a one-byte hexadecimal value preceding each record to denote "the record length in bytes". For example, if the record contains 39 bytes, z = HEX(27) since $(27)_{16} = (39)_{10}$.

record = one complete record with trailing spaces truncated, except in the first record; individual records do not overlap from one 248-byte array to the next. (Records should not exceed 128 bytes in files likely to undergo media conversion from Wang to 3740 diskettes using Wang's 3740 Diskette Compatibility Software.)

> NOTE:
>
> Following the sector containing x = HEX(F0), there must be a file trailer record written by a DATASAVE DC END statement.

APPENDIX B

INDEX TRACK LAYOUT FOR 3740 DISKETTES

Table B-1. The 3740 Index Track

Sector	Use	Initialized To
01	Reserved.	1-80 = X'40' 81-128 = X'00'
02	Reserved.	1-80 = X'40' 81-128 = X'00'
03	Reserved for system scratch use.	1-80 = X'40' 81-128 = X'00'
04	Reserved.	1-80 = X'40' 81-128 = X'00'
05	Bytes 1-5 = ERMAP, a label identifying the sector as an error map.	1-5 = C'ERMAP'
	Byte 6 is a separator and contains a blank.	6 = X'40'
	Bytes 7-8 contain blanks if no defective track exists, or the track number of the first defective physical track.	7-8 = X'40'
	Byte 9 contains a blank if no defective track exists, or zero if any defective tracks exist.	9 = X'40'
	Byte 10 is a separator and contains a blank.	10 = X'40'
	Bytes 11-12 contain blanks if one or no defective track exists, or the track number of the second defective physical track if more than one defective tracks exist.	11-12 = X'40'
	Byte 13 contains a blank if one or no defective track exists, or zero if more than one defective tracks exist.	13 = X'40'
	Byte 14 is a separator and contains a blank.	14 = X'40'
	Bytes 15-22 are reserved.	15-22 = X'40'

Table B-1 (Cont.). The 3740 Index Track

Sector	Use	Initialized To
05 (cont.)	Byte 23 is the diskette defect indicator. A blank indicates no defective records to be handled by the alternate physical record method are contained within the data portion of any data set extent on the volume. D indicates at least one such defective record exists.	23 = X'40'
	Byte 24 is the error directory indicator. A blank indicates no format or location for alternate physical record relocation has been previously specified. B or C indicates the contents of the defective physical records are relocated to a data set named ERRORSET. B indicates the addresses of the defective physical records are recorded in the error directory in the discontinuous binary format (OCHR). C indicates the addresses of the defective physical records are recorded in the error directory in the character decimal format (bCCHRR).	24 = X'40'
	Bytes 25-72 are the error directory, containing addresses of physical records having one or more defects. This field can contain up to 12 addresses in the discontinuous binary format (OCHR), or up to 8 addresses in the character decimal format (bCCHRR). The relocated records are in a data set named ERRORSET in the same sequence as the addresses in the directory. Unused bytes in the directory must contain binary zeros if byte 24 contains a B, or blanks if byte 24 contains a C.	25-72 = X'40' (256) = X'00'
	Bytes 73-80 are reserved.	73-80 = X'40'
	Bytes 81-128 are padded with binary zeros.	81-128 = X'00'
06	Reserved.	1-80 = X'40' 81-128 = X'00'
07	This sector, called the volume label, identifies the diskette's owner, security, sequence, and length of physical records.	
	Bytes 1-4 identify the sector as a volume label.	1-4 = C'VOL1'

Table B-1 (Cont.). The 3740 Index Track

Sector	Use	Initialized To
07 (cont.)	Bytes 5-10, called the volume identifier, can contain the same volume identifier (serial number) written on the diskette permanent label. The ID consists of one to six digits or letters. The first character must be in byte 5; no embedded blanks are allowed; any unused bytes to the right of the ID data must be blank. When the diskette is initialized by an IBM device, this field contains the value specified during the initialization procedure.	5-10 = C'IBMIRD'
	Byte 11 is the volume accessibility field. A blank permits access to the diskette; any nonblank character means additional qualifications are required for further access.	11 = X'40'
	Bytes 12-37 are reserved.	12-37 = X'40'
	Bytes 38-51, called the owner identifier field, are not used by all systems.	38-51 = X'40'
	Bytes 52-71 are reserved.	52-71 = X'40'
	Byte 72 is called the volume surface indicator and contains a blank.	72 = X'40'
	Byte 73 is the extent arrangement indicator and contains a blank or a P. A blank indicates there are no special constraints on the arrangement of extents, data set labels, or unallocated space on this diskette. P indicates the extents must be adjacent and must begin at track 1, sector 1; the data set labels must begin at track 0, sector 8 and must be in the same sequence as the extents they describe; all unallocated space must follow the last data set extent on the volume. If any unused space is created elsewhere, the extents must be rearranged to eliminate the space, or this field must be changed to a blank.	73 = X'40'
	Byte 74 is the special requirements indicator and contains a blank or an R. A blank indicates there are no special requirements for accessing data on this volume. An R indicates some of the data sets are recorded in a logically nonsequential manner.	74 = X'40'

Table B-1 (Cont.). The 3740 Index Track

Sector	Use	Initialized To
07 (cont.)	Byte 75, the logical recording sequence indicator, is always blank. A blank indicates the data is recorded sequentially, i.e., when a track is full, the next data record is written in sector 1 of the next higher numbered track.	75 = X'40'
	Byte 76 identifies the physical record (sector) length for tracks 1 through 76; it contains a blank, 1, or 2, where: blank = 128 bytes 1 = 256 bytes 2 = 512 bytes.	76 = X'40' (256) = C'1' (512) = C'2'
	Bytes 77-78 give the physical record (sector) code, using blanks or the characters 01 through 13 to indicate the physical sequence of the sectors. A blank or 1 indicates the sectors are physically sequential. Any other value is an increment to determine the next physical sector. Diskettes initialized on an IBM device may have a value specified during the initialization procedure.	77-78 = X'40'
	Byte 79 is reserved.	79 = X'40'
	Byte 80 is the label standard version field. W indicates IBM standard labels are on the diskette.	80 = C'W'
	Bytes 81-128 are padded with binary zeros.	81-128 = X'00'
08-26	These sectors contain the data set labels defining the data sets recorded on tracks 01 through 07. Sectors 09 through 26 are initialized as deleted records. (See Data Set Label Layout.)	See Data Set Label Layout

Note: Values given in the "Initialized To" column are for diskettes with 128 bytes per sector. Exceptions for other diskettes are noted by the number of bytes per sector in parentheses followed by the value, e.g., (256) = C'1' means the 256 bytes-per-sector diskette has a value of C'1'.

Appendix B. Index Track Layout for 3740 Diskettes

Table B-2. Data Set Label Layout

Byte-position Representation of the Index Track Data in Any Sectors from 08 through 26			Field Contents in Unused New Diskette	
Byte	Label	Description	Sector 08	Sectors 09-26
1-4	Label ID (identifier)	Label identifier for system application.	C'HDR1'	C'DDR1'
5		Reserved.	C'b'	C'b'
6-22	Data set identifier	User name for data set (1 to 17 characters). The first character must be alphabetic and in byte 6. No embedded blanks are allowed. For basic data exchange, only the first 8 characters are used. Duplicate names are not permitted on the same diskette; ERRORSET and SYSAREA are reserved for special use.	C'DATAb...b'	C'DATA09b...b' through C'DATA26b...b' (256)='b...b'
23-27	Block length	A numeric value (1-32767) specifying the maximum number of characters per block. The value must be entered at label creation. Blocks must begin on physical record boundaries. For a basic exchange data set, this field must be 1-128.	C'bb080' (256)=C'00256'	C'bb080' (256)=C'bbbbb'
28	Record attribute	Blank if records within data set are unblocked, unspanned (or if byte 44 is blank). R = blocked, spanned records. B = blocked, unspanned.	C'b'	C'b'
29-33	Beginning of extent (BOE)	The address of the data set's first sector. Bytes 29-30 give the track number, byte 31 gives the head number, and bytes 32-33 give the sector number. (Some systems use a logical record number; if so, byte 74 of the volume label contains an R.)	C'01001'	C'74001' (256)=C'bbbbb'

Table B-2 (Cont.). Data Set Label Layout

Byte-position Representation of the Index Track Data in Any Sectors from 08 through 26			Field Contents in Unused New Diskette	
Byte	Label	Description	Sector 08	Sectors 09-26
34	Physical record length	Indicates physical record length: b=128 bytes per record 1=256 bytes per record 2=512 bytes per record. The value must be the same as byte 76 of the volume label, or must be blank when byte 44 is blank.	C'b' (256)=C'1' (512)=C'2'	C'b' (512)=C'2'
35-39	End of extent (EOE)	The address of the last sector reserved for the data set, using the same format as BOE.	C'73026' (256)=C'74015' (512)=C'74008'	C'73026' (256)=C'bbbbb' (512)=C'74008'
40	Record/ block format	Contains a blank or an F and indicates fixed-length records in fixed blocks. Must be blank when byte 44 is blank.	C'b'	C'b'
41	Bypass indicator	If set to B, indicates a data set to be skipped during exchange or copy operations when transmitting or transferring the data sets on the volume. If set to blank, the data set is transferred.	C'b'	C'b'
42	Data set security	Blank indicates the data set is not secured (can be accessed). Nonblank character means restricted access (the volume accessibility indicator in sector 07, byte 11 must also be nonblank).	C'b'	C'b'
43	Write protect	P indicates the data set can be read only. A blank allows both reading and writing.	C'b'	C'b'

Table B-2 (Cont.). Data Set Label Layout

Byte-position Representation of the Index Track Data in Any Sectors from 08 through 26			Field Contents in Unused New Diskette	
Byte	Label	Description	Sector 08	Sectors 09-26
44	Exchange type indicator	Blank indicates the data set can be used for basic data exchange; any other value indicates additional label checking must be performed in order to exchange the data set.	C'b' (256,512)=C'E'	C'b' (512)=C'E'
45	Multi-volume indicator	Blank indicates the data set is wholly contained on this diskette; C indicates the data set is continued on another diskette; L indicates the last diskette on which the continued data set resides.	C'b'	C'b'
46-47	Volume sequence number	Specifies the sequence of volumes in a multivolume data set. The sequence must be consecutive, from 01 to a maximum of 99. Blanks indicate volume sequence checking is not to be performed on this volume and all subsequent volumes of a multivolume data set.	C'bb'	C'bb'
48-53	Creation date	If used, indicates the date the data set was created. The format is YYMMDD, where YY is the low-order 2 digits of the year, MM is a 2-digit representation of the month, and DD is a 2-digit representation of the day of the month. Blanks indicate the data is not significant.	C'bbbbbb'	C'bbbbbb'

Table B-2 (Cont.). Data Set Label Layout

Byte-position Representation of the Index Track Data in Any Sectors from 08 through 26			Field Contents in Unused New Diskette	
Byte	Label	Description	Sector 08	Sectors 09-26
54-57	Record length	Must be defined at label creation. Blank means the record length equals the block length defined in byte 23. (A blank in byte 44 also means record length equals block length; therefore, this field can be ignored.)	C'bbbb'	C'bbbb'
58-62	Offset to next record space	Used only in conjunction with blocked records to indicate the starting position for the next sequential record relative to the end of the last block before EOD. Blanks mean zero displacement from the next block (starts at EOD, end of data, address). A decimal value means a negative displacement.	C'bbbbb'	C'bbbbb'
63-66		Reserved.	C'bbbb'	C'bbbb'
67-72	Expiration date	If used, contains the date the data set (and its label) may be deleted. Same format as creation date. All blanks mean "expired"; all nines mean "will never expire".	C'bbbbbb'	C'bbbbbb'
73	Verify/ copy indicator	Must be blank, V, or C. Blank must be entered when the data set is created. Systems supporting verification enter a V when the data set has been verified. Systems supporting copy verification enter a C when the data has been successfully transferred to another medium (e.g., tape, transmission network). Do not enter C for partial data set copy or for null data set.	C'b'	C'b'

Table B-2 (Cont.). Data Set Label Layout

Byte-position Representation of the Index Track Data in Any Sectors from 08 through 26			Field Contents in Unused New Diskette	
Byte	Label	Description	Sector 08	Sectors 09-26
74	Data set organization	Must be a blank, an S, or D. Blank or S indicates sequential organization. D means an organization not permitting the sequential relocation method of processing defective physical records. Must be blank when byte 44 is blank.	C'b'	C'b'
75-79	End of data (EOD)	The address of the next unused sector within the data set extent, using the same format as BOE. If the address is the same as BOE, the extent contains a null data set. If the address is the next block beyond the extent (for unblocked, unspanned records), the entire extent has been used. For blocked or spanned records, this field must be used with offset to the next record space (bytes 58-62) to determine the end of the recorded data.	C'01001'	C'74001' (256)=C'bbbbb' (512)=C'75001'
80		Reserved.	C'b'	C'b'
81-128		Padded with binary zeros.	X'00'	X'00'

Note: Values given in the "Field Contents in Unused New Diskette" columns are for diskettes with 128 bytes per sector. Exceptions for other diskettes are noted by the number of bytes per sector in parentheses followed by the value, e.g., (256) = C'1' means the 256 bytes-per-sector diskette has a value of C'1'.

INDEX

To help us to provide you with the best manuals possible, please make your comments and suggestions concerning this publication on the form below. Then detach, fold, tape closed and mail to us. All comments and suggestions become the property of Wang Laboratories, Inc. For a reply, be sure to include your name and address. Your cooperation is appreciated.

700-4369C

TITLE OF MANUAL: **DISKETTE COMPATIBILITY SOFTWARE USER MANUAL**

COMMENTS:

Fold

Fold

(Please tape. Postal regulations prohibit the use of staples.)

WANG

Fold

FIRST CLASS
PERMIT NO. 16
Tewksbury, Mass.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

— POSTAGE WILL BE PAID BY —

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE
LOWELL, MASSACHUSETTS 01851

Attention: Technical Writing Department

Fold

Cut along dotted line.

Printed in U.S.A.
13-1019

International Representatives

Argentina
Bahamas
Bahrain
Bolivia
Botswana
Brazil
Canary Islands
Chile
Colombia
Costa Rica
Cyprus
Denmark
Dominican Republic
Ecuador
Egypt
El Salvador
Finland
Ghana
Greece
Guam
Guatemala
Haiti
Honduras
Iceland
India
Indonesia
Ireland
Israel
Italy
Ivory Coast
Japan
Jordan
Kenya
Korea
Kuwait
Lebanon
Liberia
Malaysia
Malta
Mexico
Morocco
New Guinea
Nicaragua
Nigeria
Norway
Paraguay
Peru
Philippines
Portugal
Qatar
Saudi Arabia
Scotland
Senegal
South Africa
Spain
Sri Lanka
Sudan
Tasmania
Thailand
Turkey
United Arab Emirates
Uruguay
Venezuela
Zimbabwe

United States

Alabama
Birmingham
Mobile

Alaska
Anchorage
Juneau

Arizona
Phoenix
Tucson

California
Anaheim
Burlingame
Culver City
Emeryville
Fountain Valley
Fresno
Los Angeles
Sacramento
San Diego
San Francisco
Santa Clara
Ventura

Colorado
Englewood

Connecticut
New Haven
Stamford
Wethersfield

District of Columbia
Washington

Florida
Coral Gables
Hialeah
Hollywood
Jacksonville
Miami
Orlando
Sarasota
Tampa

Georgia
Atlanta
Savannah

Hawaii
Honolulu
Maui

Idaho
Boise

Illinois
Arlington Heights
Chicago
Morton
Oakbrook
Park Ridge
Rock Island
Rosemont
Springfield

Indiana
Fort Wayne
Indianapolis
South Bend

Iowa
Ankeny

Kansas
Overland Park
Wichita

Kentucky
Louisville

Louisiana
Baton Rouge
Metairie

Maine
Portland

Maryland
Baltimore
Bethesda
Gaithersburg
Rockville

Massachusetts
Boston
Burlington
Chelmsford
Lawrence
Littleton
Lowell
Methuen
Tewksbury
Worcester

Michigan
Grand Rapids
Kalamazoo
Lansing
Southfield

Minnesota
Eden Prairie
Minneapolis

Mississippi
Jackson

Missouri
Creve Coeur
St. Louis

Nebraska
Omaha

Nevada
Las Vegas

New Hampshire
Manchester

New Jersey
Bloomfield
Clifton
Edison
Mountainside
Toms River

New Mexico
Albuquerque
Santa Fe

New York
Albany
Jericho
Lake Success
New York City
Rochester
Syossett
Syracuse
Tonawanda

North Carolina
Charlotte
Greensboro
Raleigh

Ohio
Akron
Cincinnati
Cleveland
Independence
Toledo
Worthington

Oklahoma
Oklahoma City
Tulsa

Oregon
Eugene
Portland
Salem

Pennsylvania
Allentown
Erie
Harrisburg
Philadelphia
Pittsburgh
State College
Wayne

Rhode Island
Providence

South Carolina
Charleston
Columbia

Tennessee
Chattanooga
Knoxville
Memphis
Nashville

Texas
Austin
Dallas
El Paso
Houston
San Antonio

Utah
Salt Lake City

Virginia
Newport News
Norfolk
Richmond
Rosslyn
Springfield

Washington
Richland
Seattle
Spokane

Wisconsin
Appleton
Brookfield
Green Bay
Madison
Wauwatosa

International Offices

Australia
Wang Computer Pty., Ltd.
Adelaide, S.A.
Brisbane, Qld.
Canberra, A.C.T.
Perth, W.A.
South Melbourne, Vic 3
Sydney, NSW

Austria
Wang Gesellschaft, m.b.H.
Vienna

Belgium
Wang Europe, S.A.
Brussels
Erpe-Mere

Canada
Wang Canada Ltd.
Burlington, Ontario
Burnaby, B.C.
Calgary, Alberta
Don Mills, Ontario
Edmonton, Alberta
Halifax, Nova Scotia
Hamilton, Ontario
Montreal, Quebec
Ottawa, Ontario
Quebec City, Quebec
Toronto, Ontario
Victoria, B.C.
Winnipeg, Manitoba

China
Wang Industrial Co., Ltd.
Taipei

Wang Laboratories, Ltd.
Taipei

France
Wang France S.A.R.L.
Paris
Bordeaux
Lille
Lyon
Marseilles
Nantes
Nice
Rouen
Strasbourg

Great Britain
Wang (U.K.) Ltd.
Richmond
Birmingham
London
Manchester

Hong Kong
Wang Pacific Ltd.
Hong Kong

Japan
Wang Computer Ltd.
Tokyo

Netherlands
Wang Nederland B.V.
IJsselstein
Groningen

New Zealand
Wang Computer Ltd.
Auckland
Christchurch
Wellington

Panama
Wang de Panama (CPEC) S.A.
Panama City

Puerto Rico
Wang Computadoras, Inc.
Hato Rey

Singapore
Wang Computer (Pte) Ltd.
Singapore

Sweden
Wang Skandinaviska AB
Stockholm
Gothenburg
Malmö

Switzerland
Wang A.G.
Zurich
Basel
Bern
Geneva
Lausanne
St. Gallen

Wang Trading A.G.
Zug

West Germany
Wang Deutschland, GmbH
Frankfurt
Berlin
Cologne
Düsseldorf
Essen
Freiburg
Hamburg
Hannover
Kassel
Mannheim
Munich
Nürnberg
Saarbrücken
Stuttgart

WANG

ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851, TEL. (617) 459-5000, TWX 710 343-6769, TELEX 94-7421

Printed in U.S.A.
700-4369C
2-82-1M