

WANG

SYSTEM 2200  
GENERAL UTILITIES  
VOLUME II

# SYSTEM 2200





# **2200**

## **General Utilities**

### **Volume II**

© Wang Laboratories, Inc., 1975



LABORATORIES, INC.

---

B36 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876. TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

**Warranty Disclaimer and Consequential Damages**

The programming staff of Wang Laboratories, Inc. has taken due care to prepare this program package, including research, development, and testing to ascertain its effectiveness. Wang Laboratories, Inc. and its subsidiaries make no expressed or implied warranty of any kind with regard to this program package and its related material. In no event shall Wang Laboratories, Inc. or its subsidiaries be liable for incidental or consequential damages in connection with or arising out of the furnishing, performance or use of any of its programs.

**Proprietary Rights**

This program package is a company proprietary item. Tape cassettes of programs may not be reproduced without the written consent of Wang Laboratories, Inc.

**Payment Address**

The fee for a package is to be remitted directly to Wang Laboratories, Inc. at 836 North Street, Tewksbury, Massachusetts 01876. No agent or representative is charged with the authority to accept payment on behalf of the administrative offices of Wang Laboratories, Inc.



LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851 4111, TWX 710 343 6769, TELEX 94 7421

## HOW TO USE THIS MANUAL

This manual provides descriptions and operating instructions for a number of utility routines that have been prepared for users of the Wang System 2200.

The manual has been organized in the same order as the programs have been recorded on tape.

### DISK UTILITIES (one System tape)

SYSTEM TAPE: 15 files: tape # 701-0236

<u>PROGRAM NO.</u>	<u>NAME</u>	<u>NO. FILES</u>	<u>CALL</u>
1	Compression	2	"COMPRESS"
2	Decompress & Cross Reference	2	"D&C/XREF"
3	Listing	2	"LIST"
4	Tape & Disk Dump	1	"T/D-DUMP"
5	Disk Sort	5	"DISKSORT"
6	Disk & Display Utilities	1	"DISKUTIL"
7	Scratch Volume Initialization	1	"SCRATCH"
8	Date Routines	1	"DATE"

### TAPE UTILITIES (one System tape)

SYSTEM TAPE: 17 files Tape #701-0235

<u>PROGRAM NO.</u>	<u>NAME</u>	<u>NO. FILES</u>	<u>CALL</u>
1	Compression	2	"COMPRESS"
2	Decompress & Cross Reference	2	"D&C/XREF"
3	Listing	2	"LIST"
4	Tape & Disk Dump	1	"T/D-DUMP"
5	Tape Sort	7	"TAPESORT"
6	Tape & Display Utilities	1	"TAPEUTIL"
7	Scratch Volume Initialization	1	"SCRATCH"
8	Date Routines	1	"DATE"

## TABLE OF CONTENTS

		Page
1.0	COMPRESSION Program	1
1.1	Introduction	1
1.2	Step 1: General Description of the COMPRESSION Program	1
1.3	Step 2: Initialization of COMPRESSION Program Tape; Selecting Options	3
1.4	Step 2A: Executing Option 1 - Disk to Disk	3
1.5	Step 2B: Executing Option 2 - Disk to Tape	5
1.6	Step 2C: Executing Option 3 - Tape to Disk	6
1.7	Step 2D: Executing Option 4 - Tape to Tape	7
2.0	DECOMPRESS and CROSS REFERENCE Program	8
2.1	Introduction	8
2.2	Decompress and Cross Reference Setup Procedure	8
2.3	Decompress and Cross Reference Tape Input	10
2.4	Decompress and Cross Reference Disk Input	13
2.5	Processing Phase	17
2.6	Error Recovery for Processing Module	17
2.7	Error Recovery for Execution Phase	18
3.0	LISTING Program	19
3.1	Introduction	19
3.2	Step 1: Initialization	19
3.3	Tape Input	20
3.4	Disk Input	22
3.5	Processing Phase	26
3.6	Error Recovery for Processing Module	27
3.7	Error Recovery for Execution Phase	27
3.8	Error Recovery Operation	27
4.0	DISK and TAPE DUMP	28
4.1	Introduction	28
4.2	Step 1: Initialization	28
4.3	Step 2: Specifying Input Device	29
4.4	Step 3A: Using a Tape Cassette as Input Device	29
4.5	Step 3B: Using a Disk as Input Device	30
4.6	Step 4: Execution of the DUMP Program	31
5.0	DISK SORT	38
5.1	Introduction	38
5.1.1	Terminology	38
5.1.2	General System Overview	39
5.1.3	The Disk Sort System	40
5.1.4	Restrictions and Conventions	40
5.1.5	The 'EXEC' Instruction	42
5.1.6	Machine Requirements	42
5.2	Operating Instructions	45
5.3	Error Messages	50

## TABLE OF CONTENTS

	Page
6.0 TAPE SORT	54
6.1 Definition "SORT"	54
6.2 Glossary	54
6.3 General	55
6.4 Tape Sort	55
6.5 Operating Procedures	58
6.6 Error Messages	64
7.0 TAPE/DISK & DISPLAY UTILITIES	67
7.1 Introduction	67
7.2 Open Output File '240	68
7.3 End of Volume/File Output '241	69
7.4 Accept Alpha Data (with Message) '243	69
7.5 Accept Alpha Data (no message) '244	70
7.6 Accept Numeric Data (with message) '245	70
7.7 Accept Numeric Data (no message) '246	71
7.8 Position Cursor '248	71
7.9 Open Input File '250	71
7.10 Close Input Volume '251	72
7.11 Operator Wait '254	72
7.12 Variables	73
8.0 SCRATCH VOLUME INITIALIZATION	74
8.1 General	74
9.0 DATE ROUTINES	75
9.1 Introduction	75
9.2 Two Digit Pack	75
9.3 Numeric Unpack	76
9.4 Convert Character Date to Month-Day-Year	77
9.5 Convert Month-Day-Year to Year-Days	78
9.6 Convert Year-Day to Julian Date	78
9.7 Convert Julian Date to Year-Days	79
9.8 Convert Year-Days to Month-Day-Year	79
9.9 Convert Month-Day-Year to Character Form	80
9.10 Convert from Year-Day to Days	80
9.11 Adjust Year-Days to Normal Form	81
9.12 Accept Date	82
9.13 Variables	82





## 1.0

### COMPRESSION PROGRAM

#### 1.1 INTRODUCTION

The purpose of the compression program is to compress source programs on either a disk or tape and save the compressed version on either disk or tape. In compressing a program, all REM statements except the first are omitted and statement lines are compressed by removing the statement line number and inserting colons between statements.

You can choose the maximum length of any compressed statement as 180 bytes or 256 bytes. Certain line numbers (i.e., those that are referenced in the program) are not omitted in the compressed version of the program. The compressed version of a program allows for faster execution of programs, plus a large saving of memory in the System 2200B.

NOTE:

The last statement in the program to be compressed must be REM\$; no space can be entered by the user between REM and \$ or an error results. If you try to compress a program that does not have REM\$ as its last statement, an error results.

The following equipment is needed to operate the compression program: a minimum 12K System 2200B, a CRT unit, a keyboard, a cassette drive(s) and/or disk drive(s), the compression program tape, and this set of operating instructions.

#### 1.2 STEP 1: GENERAL DESCRIPTION OF THE COMPRESSION PROGRAM

When the compression program is executed, various options for input and output are shown; you are to choose one. These options, as well as detailed operating instructions for each option, are contained in the remainder of this instruction sheet (Step 2).

NOTE:

Any program to be compressed cannot contain statements branching to REM statements, as all REM statements except the first are deleted.

A program can, however, contain branching to a multi-statement line, which has a REM statement as one of its statements. For example, the following statements:

```
5 REM GENERAL LEDGER
10 GO TO 200
200 REM RECORD FILE: A=A+1
```

are compressed as follows:

- a) Statement 5 remains in the compressed version as it is the first REM statement.
- b) Statement 10 is compressed and becomes part of line 5: 5 REM GENERAL LEDGER: GO TO 200.
- c) Statement 200 remains, but the REM statement is deleted: 200 A=A+1.

After you have specified an option, the CRT displays various questions for each option, asking specific information concerning the chosen option. One of the last questions shown on the CRT before the System 2200B begins compressing the desired programs asks you to input 1 for 256 bytes, or 2 for 180 bytes. This question refers to the number of bytes you desire per statement line.

Choosing 256 bytes per statement line allows for the maximum compression of the program; however, after the program is compressed, you cannot edit any statement line that exceeds 192 bytes. The maximum size of a statement line that can be edited is 192 bytes.

Choosing 180 bytes per statement line allows you the option of editing any desired line after the program is compressed. This can be done by using the EDIT key to put the System 2200B into the Edit Mode; choosing 180 bytes per line also provides for adding approximately 12 characters.

If you should add any more than 12 characters (total = 180 + 12 = 192), then the statement cannot be edited again.

When you have input all required information, the compression program begins compressing the source program by building a protect table. During the building of this protect table, the compression program reviews every statement line in the source program and checks for branching (i.e., IF THEN, GO TO, PRINTUSING, etc.). If any branching occurs, the compression program does not take out the statement number. All REM statements except the first are deleted, and any statement numbers that do not involve branching are deleted. However, statement numbers that are the start of a new 180 or 256 byte statement line are not deleted. Program statements are thus printed, separated only by colons.

When the compression program has finished making this protect table, the CRT displays the current program number (i.e., 1, 2, ...), the number of blocks read from the source program, the number of blocks written into the compressed program, and the percent of blocks read. This display remains on the CRT until the compression is complete and the compressed program has been copied onto either a disk or tape.

The last message printed by the compression program is STOP END OF PROGRAM.

If you are finished using the compression program, simply remove your tape(s) and/or disk(s) from their respective drives.

### 1.3 STEP 2: INITIALIZATION OF COMPRESSION PROGRAM TAPE; SELECTING OPTIONS

Place the compression tape in the main tape drive (address code = 10A) and make sure it is rewound.

Key	CLEAR, RETURN(EXEC)
	LOAD "COMPRESS", RETURN(EXEC)
	RUN, RETURN(EXEC)

After loading the logic table, the CRT displays the options available with the compression program:

1. DISK TO DISK
2. DISK TO TAPE
3. TAPE TO DISK
4. TAPE TO TAPE

**NOTE:**

If output is disk, the compression program will verify output disk before beginning compression of source.

Key in the desired option number, followed by RETURN(EXEC). Depending on the option specified, proceed to Step 2A, B, C or D.

### 1.4 STEP 2A: EXECUTING OPTION 1 - DISK TO DISK

After selecting option 1, the compression program asks for the disk working order:

1. FIXED TO REMOVABLE
2. REMOVABLE TO FIXED

The two disk addresses used with this compression tape are 310 and B10. The words fixed and removable refer to disk drives on all four disk units: Models 2230, 2240, 2242 and 2243. The following table illustrates which disk drive on each Model is associated with fixed and which drive is associated with removable.

ADDRESS CODE	RELATED DISK DRIVES			
	MODEL 2230	MODEL 2240	MODEL 2242	MODEL 2243
310 (FIXED) B10 (REMOVABLE)	Fixed Removable	Left Right	Only Drive Illegal	Left Middle

NOTE:

The Model 2242 cannot be used to directly compress programs from disk to disk as the Model 2242 has only a single removable disk drive.

Also, the right unit on the Model 2243 is not used in this compression program.

After you have input the disk working order, the next display asks you to mount the disk(s). Place the disk containing the source programs into the selected disk drive; insert the disk which is to contain the compressed programs into the remaining disk drive.

TOUCH RETURN (EXEC).

The compression program now verifies the output disk. If there are errors in any of the sectors, start over with new disk. The compression program now asks specific questions concerning the number of programs to be compressed, and the names of the source programs.

NOTE:

A program on the disk cannot be compressed unless it is cataloged. The disk which is to contain the compressed program must have sufficient room for the compressed program in the catalog area.

The last question to be answered before compression begins concerns the number of bytes per statement line desired in the compressed program. Enter 1 for 256 bytes per statement line, or 2 for 180 bytes (see Step 1 for greater detail on entering the number of bytes per statement line).

The System 2200B now begins compressing the source programs specified and copies the compressed programs to the other disk. Step 1 discusses, in detail, the execution of the compression program.

## 1.5 STEP 2B: EXECUTING OPTION 2 - DISK TO TAPE

After selecting option 2, the compression program asks for the desired disk option:

1. FIXED DISK
2. REMOVABLE DISK

The words fixed and removable refer to disk drives on all four disk units: Models 2230, 2240, 2242 and 2243. The table on page 4 illustrates which disk drive on each Model is associated with fixed, and which disk drive is associated with removable.

When the compression program asks you to mount the disk(s), place the disk containing the source programs in the specified drive.

Touch RETURN(EXEC).

The compression program continues by asking for the number of source programs on the disk and their names.

NOTE:

A program on disk cannot be compressed unless it is cataloged.

The last question asked by the compression program is concerned with the number of bytes per statement. Enter 1 for 256 bytes per statement line, or 2 for 180 bytes (see Step 1 for greater detail on entering the number of bytes per statement line).

The second module of the compression program is now loaded into memory. The program now asks you to mount the tape.

NOTE:

When compressing a program from the disk to a tape, the compression program automatically selects address code 10A for the tape.

The system 2200B now begins compressing the source programs on the disk and copies them onto the tape. Step 1 illustrates, in detail, the execution of the compression program.

## 1.6 STEP 2C: EXECUTING OPTION 3 - TAPE TO DISK

After selecting option 3, the compression program asks for the disk option:

1. FIXED DISK
2. REMOVABLE DISK

The words fixed and removable refer to disk drives on all four disk units: Models 2230, 2240, 2242, and 2243. The table shown on page 4 illustrates which disk drive on each unit is associated with fixed, and which drive is associated with removable.

The next CRT display asks you to mount the disk(s). Place the disk which is to contain the compressed programs in the specified disk drive.

NOTE:

The output disk is verified before compression begins.

The compression program continues by asking specific information about your programs: the number of programs to be compressed.

NOTE:

A newly compressed program on the disk must have a name (i.e., it must be cataloged). Therefore, the disk which is to contain the compressed programs must have a catalog index and enough room for the compressed program in the catalog area.

The last question to be answered concerns the number of bytes per statement line. Enter 1 for 256 bytes per statement line, or 2 for 180 bytes (see Step 1 for greater detail on entering the number of bytes per statement line).

The second module of the program is now loaded into memory. The program now asks you to mount the tape.

NOTE:

The compression program automatically selects address code 10A as the tape drive.

The System 2200B now begins compressing the source programs on the tape and copies them to the disk. Step 1 illustrates, in detail, the execution of the compression program.

## 1.7 STEP 2D: EXECUTING OPTION 4 - TAPE TO TAPE

1. After selecting option 4, the compression program asks you the number of programs to be compressed.
2. The compressed program continues by asking whether you want 256 bytes per statement line (enter 1), or 180 bytes per statement line (enter 2). Refer to Step 1 for greater details on entering the number of bytes per statement line. The second module of the program is loaded into memory.
3. Remove the compression program from the main tape drive (address code = 10A), and insert the tape containing the source programs to be compressed. Place the tape which is to contain the compressed programs in the second tape drive (address code = 10B).

NOTE:

When compressing programs from tape to tape, the compression program automatically selects the source tape drive as 10A, and the tape drive which is to contain the compressed programs as 10B.

4. The System 2200B begins compressing the source programs on tape and copies them to the other tape. Step 1 illustrates, in detail, the execution of the compression program.

## 2.0

### DECOMPRESS AND CROSS REFERENCE PROGRAM

#### 2.1 INTRODUCTION:

The following equipment is needed to operate the D&C (decompress and cross reference) program: a minimum 12K Wang System 2200B, a CRT unit, a keyboard, a cassette drive, a disk drive (if input is to be obtained from a disk), an output device, and this set of operating instructions.

The D&C program takes any program, on tape cassette or disk, decompresses and/or cross references it. The results of the D&C program can be output on a number of peripherals. Decompressing and cross referencing are two major ways of spotting "bugs" in a program.

When a program is decompressed, each statement is printed on a separate line. Multi-statement lines are separated (at the colon) and each single statement is printed on a separate line. This allows you to view each statement line for errors and to follow program flow more easily.

When a program is cross referenced, the variables in the specified program are printed out alphabetically, with a list of line numbers that reference each variable, and a list of line numbers are printed with a list of line numbers which reference each. Throughout the remainder of these instructions, D&C is used to indicate decompress and cross reference.

#### 2.2 DECOMPRESS AND CROSS REFERENCE SETUP PROCEDURE

1. To initialize the program for your system, go to Step 2 of this section.

1a. To run the Program Tape Utilities System

```
Key    CLEAR, RETURN(EXEC)
        LOAD "D&C/XREF", RETURN(EXEC)
        RUN, RETURN(EXEC)
```

Go to the appropriate input section.

2. To load the second module:

```
Key    CLEAR
        RETURN(EXEC)
        LOAD "D&C10205"
        RETURN(EXEC)
        LISTS
        RETURN(EXEC)
```

For maximum effectiveness, you should change the array size of your program to correspond to the core (k) size of your system.



NOTE:

If you wish to change the array size and rerecord the second module of the D&C program, be sure you have a backup of the program tape (see General Utilities III for Tape to Tape Copy).

Do not set the protection tab on the program tape at this time if you must change the size of the array for the variable and cross reference table. You should set the tab after you have changed the size of the array and saved it on the program tape.

3. To change the program, do the following:

Key in statement line 130 again, using the desired value for L4. Also, V2\$() is the variable representing the entire array; change the dimensions of this variable to correspond to L4 (V2\$() is also defined in statement line 130).

Size of System (k)	12	16	20	24	28	32
Variables						
L4	3	7	11	15	16	16
V2\$(L4,255)4	V2\$(3,255)4	V2\$(7,255)4	V2\$(11,255)4	V2\$(15,255)4	V2\$(16,255)4	V2\$(16,255)4

4. After changing line 130,

Key BACKSPACE 1F  
SAVE "D&C10205"

5. Go to Step 1.

2.3 DECOMPRESS AND CROSS REFERENCE TAPE INPUT

1. INSTRUCTION

KEY IN THE DATE  
?-----

2. INSTRUCTION

INPUT DEVICE TYPE 1-TAPE OR 2-DISK.      DEFAULT=2  
?-/

NOTE:

To use the defaults, just key RETURN(EXEC).

3. INSTRUCTION

KEY IN THE TAPE UNIT NUMBER, SEE TABLE BELOW.      DEFAULT=1  
?-/

\*\*\* ADDRESS TABLE \*\*\*

1 - 10A	4 - 10D
2 - 10B	5 - 10E
3 - 10C	6 - 10F

4. INSTRUCTION

ENTER THE NUMBER OF PROGRAM AND DATA FILES TO SKIP.      DEFAULT=0  
?--/

5. INSTRUCTION

FORM OF OUTPUT 1-HARD COPY 2-DISPLAY.      DEFAULT=1  
?-/

6. INSTRUCTION

IF USING 2-DISPLAY, GO TO STEP 14.

7. INSTRUCTION

ENTER PRINTER UNIT NUMBER, SEE TABLE BELOW.      DEFAULT=5  
?-/

\*\*\* ADDRESS TABLE \*\*\*

1 - 211	4 - 214
2 - 212	5 - 215
3 - 213	6 - 216

8. INSTRUCTION

READY PRINTER  
KEY RETURN(EXEC) TO RESUME  
?-

ERROR MESSAGE "PRINTER IS NOT READY", RECOVERY PROCEDURE: TURN ON PRINTER

9. INSTRUCTION

NUMBER OF SPACES FOR THE MARGIN (1-30).      DEFAULT=10  
?--/

10. INSTRUCTION

ENTER THE NUMBER OF LINES PER PAGE (10-60).      DEFAULT=60  
?--/

11. INSTRUCTION

ENTER THE LINE LENGTH INCLUDING MARGIN (50-130).      DEFAULT=80  
?---/

NOTE:

The legal range for each parameter above is provided on the CRT screen with each question concerning the hard copy output. All ranges shown, with the exception of the range shown for the line length, apply to every hard copy device operable with this program. The maximum entry shown for the line length is 130 characters. Be sure to insert a line length compatible with whatever output device you are using.

Below is the format of a line with length = 60 and margin of 10.

MARGIN	
10 spaces	printed line length = 50.

Total line length = 60.

12. INSTRUCTION

ENTER THE NUMBER OF PROGRAM FILES TO BE PROCESSED.  
?--/

13. INSTRUCTION

ENTER OPTION, SEE TABLE BELOW.      DEFAULT=3  
?-/

\*\*\*\*\* OPTIONS AVAILABLE \*\*\*\*\*

- 1 - CROSS REFERENCE
- 2 - DECOMPRESSED LISTING
- 3 - DECOMPRESS AND CROSS REFERENCE

14. INSTRUCTION

ARE THE PARAMETERS OK? (Y/N)  
?-/

15. INSTRUCTION

IF "N" IS ENTERED, GO TO STEP 1.

16. INSTRUCTION

Go to Processing Phase.

2.4 DECOMPRESS AND CROSS REFERENCE DISK INPUT

1. INSTRUCTION

KEY IN THE DATE

?-----

2. INSTRUCTION

INPUT DEVICE TYPE 1-TAPE OR 2-DISK.      DEFAULT=2

?-/

NOTE:  
To use the defaults, just key RETURN(EXEC).

3. INSTRUCTION

ENTER PLATTER 1-FIXED OR 2-REMOVABLE.      DEFAULT=1

?-/

NOTE:  
The words fixed and removable refer to disk drives on all five disk units: Models 2230, 2260, 2240, 2242, and 2243. The following table illustrates which disk drive on each Model is associated with fixed, and which drive is associated with removable.

Address Codes	Related Disk Drives			
	Model 2230	Model 2240	Model 2242	Model 2243
(FIXED) 310, 320, 330, 340	Fixed	Left	Only Drive	Left
(REMOVABLE) B10, B20, B30, B40, B50, B60	Removable	Right	Illegal	Middle
(FIXED) 350, 360	Fixed	Left	Only Drive	Right

4. INSTRUCTION

KEY IN THE DISK UNIT NUMBER, SEE TABLE BELOW.      DEFAULT=1  
?-/

USE THIS TABLE FOR FIXED DISK.  *** ADDRESS TABLE *** 1 - 310            4 - 340 2 - 320            5 - 350 3 - 330            6 - 360	USE THIS TABLE FOR REMOVABLE DISK.  *** ADDRESS TABLE *** 1 - B10            4 - B40 2 - B20            5 - B50 3 - B30            6 - B60
--	--

5. INSTRUCTION

MOUNT PROGRAM DISK IN UNIT 1  
KEY RETURN(EXEC) TO RESUME.  
?-

6. INSTRUCTION

FORM OF OUTPUT 1-HARD COPY 2-DISPLAY.      DEFAULT=1  
?-/

7. INSTRUCTION

IF USING 2-DISPLAY, GO TO STEP 13.

8. INSTRUCTION

ENTER PRINTER UNIT NUMBER, SEE TABLE BELOW.      DEFAULT=5  
?-/

*** ADDRESS TABLE *** 1 - 211            4 - 214 2 - 212            5 - 215
---

9. INSTRUCTION

READY PRINTER.  
KEY RETURN(EXEC) TO RESUME  
?-  
ERROR MESSAGE "PRINTER IS NOT READY", RECOVERY: TURN ON PRINTER

10. INSTRUCTION

NUMBER OF SPACES FOR THE MARGIN (1-30).      DEFAULT=10  
?--/

11. INSTRUCTION

ENTER THE NUMBER OF LINES PER PAGE (10-60).      DEFAULT=60  
?--/

12. INSTRUCTION

ENTER THE LINE LENGTH INCLUDING MARGIN (50-130).      DEFAULT=80  
?---/

NOTE:

The legal range for each parameter above is provided on the CRT screen with each question concerning the hard copy output. All ranges shown, with the exception of the range shown for the line length, apply to every hard copy device operable with this program. The maximum entry shown for the line length is 130 characters. Be sure to insert a line length compatible with whatever output device you are using.

Below is the format of a line with length = 60, and margin of 10.

MARGIN	
10 SPACES	PRINTED LINE LENGTH = 50
TOTAL LINE LENGTH = 60	

13. INSTRUCTION

DO YOU WANT ALL FILES TO BE PROCESSED? Y/N  
?-/

NOTE:

If the ALL option is taken, the files will be processed in the order they appear when a LIST DC F command is executed.

14. INSTRUCTION

IF THE ALL OPTION IS TAKEN, GO TO STEP 18.

15. INSTRUCTION

ENTER THE NUMBER OF PROGRAM FILES TO BE PROCESSED.  
?--/

16. INSTRUCTION

ENTER THE NAME OF FILE 1

ERROR MESSAGE "RE-ENTER PROGRAM ----- NOT AN ACTIVE PROGRAM FILE", IF FILE NAME IS NOT AN ACTIVE FILE ON DISK.

17. INSTRUCTION

REPEAT STEP 16 UNTIL ALL FILE NAMES HAVE BEEN ENTERED. AFTER THE LAST FILE NAME HAS BEEN ENTERED, THE FOLLOWING MESSAGE IS DISPLAYED:

KEY RETURN(EXEC) TO RESUME.  
?-

18. INSTRUCTION

ENTER OPTION; SEE TABLE BELOW. DEFAULT=3.  
?-/

\*\*\*\*\* OPTIONS AVAILABLE \*\*\*\*\*  
1 - CROSS REFERENCE  
2 - DECOMPRESSED LISTING  
3 - DECOMPRESS AND CROSS REFERENCE

19. INSTRUCTION

ARE THE PARAMETERS OK? (Y/N)  
?-/

20. INSTRUCTION

IF "N" IS ENTERED, GO TO STEP 1.

21. INSTRUCTION

GO to Processing Phase.



2.5 PROCESSING PHASE

1. MESSAGE

INITIALIZING TABLES FOR NEXT MODULE

2. MESSAGE

LOADING PROCESSOR

3. INSTRUCTION

IF DISK INPUT, SKIP STEP 4.

4. INSTRUCTION

MOUNT INPUT TAPE IN UNIT #

KEY RETURN(EXEC) TO RESUME?

5. MESSAGE

PROGRAM EXECUTION CONTINUES UNTIL ALL FILES HAVE BEEN PROCESSED OR UNTIL THE PROGRAM INTERRUPT IS USED.

2.6 ERROR RECOVERY FOR PROCESSING MODULE

S.F. 0 - RE-START LISTING AT THE BEGINNING OF THE CURRENT FILE

S.F. 15 - RE-PRINT CROSS REFERENCE TABLES

KEY "H" TO INTERRUPT EXECUTION AT THE END OF THE CURRENT FILE

## 2.7 ERROR RECOVERY FOR EXECUTION PHASE

THERE ARE TWO ERROR RECOVERY ROUTINES AND ONE INTERRUPTION ROUTINE IN THE SECOND MODULE OF THIS PROGRAM. THE ERROR RECOVERY ROUTINES ALLOW THE USER TO RESTART THE PROGRAM AT THE BEGINNING OF THE CURRENT FILE OR TO REPRINT THE CROSS REFERENCE TABLES. SHOULD THE PRINTER RUN OUT OF PAPER THESE ROUTINES CAN BE USED.

THE INTERRUPT ROUTINE ALLOWS THE USER TO HALT THE EXECUTION OF THE PROGRAM AT THE END OF THE CURRENT FILE AND CONTINUE EXECUTION BY KEYING RETURN(EXEC). THIS ALLOWS THE USER TO STOP THE PROGRAM TO MOUNT ADDITIONAL PAPER IN THE PRINTER WITHOUT UPSETTING ANY OF THE INTERNAL COUNTERS AND POINTERS OR DISTURBING THE DISPLAY.

### ERROR RECOVERY OPERATION

TO RESTART THE CURRENT FILE AT THE BEGINNING WITH THE SPECIFIED OPTION, KEY SPECIAL FUNCTION 0.

TO REPRINT THE CURRENT CROSS REFERENCE TABLES, KEY SPECIAL FUNCTION 15.

TO INTERRUPT PROGRAM EXECUTION, KEY THE CAPITAL LETTER H AND THE PROGRAM WILL STOP AT THE END OF THE CURRENT FILE.

## 3.0

### LISTING PROGRAM

#### 3.1 INTRODUCTION:

The following equipment is necessary to operate the Listing program: a minimum 8K System 2200B, a CRT Unit, a keyboard, a cassette drive, a disk drive (if input is to be obtained from disk), the Listing program, an output device if hardcopy is desired, and this set of operating instructions.

The purpose of the Listing program is to allow both source and compressed programs, on tape cassette or disk, to be listed in a decompressed format. In a decompressed listing of a program, each statement is printed on a separate line. Multi-statement lines are separated at the colon(s) and each single statement is printed on a separate line. This type of program listing allows you to view each line and to follow the program flow more easily.

The results of the Listing program can be printed out on a number of peripherals.

The remainder of this set of instructions contains operating instructions for the Listing program and explanations.

#### 3.2 STEP 1: INITIALIZATION

Place the General Utilities II program tape in the main tape drive (address code = 10A), and make sure it is rewound.

Key	CLEAR, RETURN(EXEC)
	LOAD "LIST", RETURN(EXEC)
	RUN, RETURN(EXEC)

This procedure clears the memory of the System 2200B, loads the first module of the Listing program and begins execution of the first module.

3.3 TAPE INPUT

1. INSTRUCTION

KEY IN THE DATE  
?-----

2. INSTRUCTION

INPUT DEVICE TYPE 1-TAPE OR 2-DISK.      DEFAULT=2  
?-/

NOTE:  TO USE THE DEFAULTS, JUST KEY RETURN(EXEC).
--

3. INSTRUCTION

KEY IN THE TAPE UNIT NUMBER, SEE TABLE BELOW.      DEFAULT=1  
?-/

*** ADDRESS TABLE ***	
1 - 10A	4 - 10D
2 - 10B	5 - 10E
3 - 10C	6 - 10F

4. INSTRUCTION

ENTER THE NUMBER OF PROGRAM AND DATA FILES TO SKIP.      DEFAULT=0  
?--/

5. INSTRUCTION

FORM OF OUTPUT 1-HARD COPY 2-DISPLAY.      DEFAULT=1  
?-/

6. INSTRUCTION

IF USING 2-DISPLAY, GO TO STEP 14.

7. INSTRUCTION

ENTER PRINTER UNIT NUMBER, SEE TABLE BELOW.      DEFAULT=5  
?-/

*** ADDRESS TABLE ***	
1 - 211	4 - 214
2 - 212	5 - 215
3 - 213	6 - 216

8. INSTRUCTION

READY PRINTER  
KEY RETURN(EXEC) TO RESUME  
?-  
ERROR MESSAGE "PRINTER IS NOT READY", RECOVERY: TURN ON PRINTER

9. INSTRUCTION

NUMBER OF SPACES FOR THE MARGIN (1-30).      DEFAULT=10.  
?--/

10. INSTRUCTION

ENTER THE NUMBER OF LINES PER PAGE (10-60).      DEFAULT=60.  
?--/

11. INSTRUCTION

ENTER THE LINE LENGTH INCLUDING MARGIN (50-130).      DEFAULT=80.  
?---/

NOTE:

The legal range for each parameter above is provided on the CRT screen with each question concerning the hard copy output. All ranges shown, with the exception of the range shown for the line length, apply to every hard copy device operable with this program. The maximum entry shown for the line length is 130 characters. Be sure to insert a line length compatible with whatever output device you are using.

Below is the format of a line with length = 60, and margin of 10.

MARGIN	
10 SPACES	PRINTED LINE LENGTH = 50.

TOTAL LINE LENGTH = 60.

12. INSTRUCTION

ENTER THE NUMBER OF PROGRAM FILES TO BE PROCESSED.  
?--/

13. INSTRUCTION

ARE THE PARAMETERS OK? (Y/N)  
?-/

14. INSTRUCTION

IF "N" IS ENTERED, GO TO STEP 1.

15. INSTRUCTION

GO to Processing Phase.

3.4 DISK INPUT

1. INSTRUCTION

KEY IN THE DATE

?-----

2. INSTRUCTION

INPUT DEVICE TYPE 1-TAPE OR 2-DISK.      DEFAULT=2

NOTE:  
TO USE THE DEFAULTS, JUST KEY RETURN(EXEC).

3. INSTRUCTION

ENTER PLATTER 1-FIXED OR 2-REMOVABLE.      DEFAULT=1

?-/

NOTE:  
The words fixed and removable refer to disk drives on all five disk units: Models 2230, 2260, 2240, 2242, and 2243. The following table illustrates which disk drive on each Model is associated with fixed, and which drive is associated with removable.

Address Codes	Related Disk Drives			
	Model 2230	Model 2240	Model 2242	Model 2243
(FIXED) 310, 320, 330, 340	Fixed	Left	Only Drive	Left
(REMOVABLE) B10, B20, B30, B40, B50, B60	Removable	Right	Illegal	Middle
(FIXED) 350, 360	Fixed	Left	Only Drive	Right

4. INSTRUCTION

INPUT DEVICE TYPE DISK

KEY IN THE DISK UNIT NUMBER, SEE TABLE BELOW.      DEFAULT=1

?-/

USE THIS TABLE FOR FIXED DISK.	USE THIS TABLE FOR REMOVABLE DISK.
*** ADDRESS TABLE *** 1 - 310            4 - 340 2 - 320            5 - 350 3 - 330            6 - 360	*** ADDRESS TABLE *** 1 - B10            4 - B40 2 - B20            5 - B50 3 - B30            6 - B60

5. INSTRUCTION

MOUNT PROGRAM DISK IN UNIT 1  
 KEY RETURN(EXEC) TO RESUME.  
 ?-

6. INSTRUCTION

FORM OF OUTPUT 1-HARD COPY 2-DISPLAY.          DEFAULT=1  
 ?-/

7. INSTRUCTION

IF USING 2-DISPLAY, GO TO STEP 13.

8. INSTRUCTION

ENTER PRINTER UNIT NUMBER, SEE TABLE BELOW.          DEFAULT=5  
 ?-/

*** ADDRESS TABLE ***
1 - 211            4 - 214
2 - 212            5 - 215
3 - 213            6 - 216

9. INSTRUCTION

READY PRINTER  
 KEY RETURN(EXEC) TO RESUME  
 ?  
 ERROR MESSAGE "PRINTER IS NOT READY", RECOVERY: TURN ON PRINTER.

10. INSTRUCTION

NUMBER OF SPACES FOR THE MARGIN (1-30).          DEFAULT=10  
 ?--/

11. INSTRUCTION

ENTER THE NUMBER OF LINES PER PAGE (10-60).          DEFAULT=60  
 ?--/

12. INSTRUCTION

ENTER THE LINE LENGTH INCLUDING MARGIN (50-130).      DEFAULT=80  
?---/

NOTE:

The legal range for each parameter above is provided on the CRT screen with each question concerning the hard copy output. All ranges shown, with the exception of the range shown for the line length, apply to every hard copy device operable with this program. The maximum entry shown for the line length is 130 characters. Be sure to insert a line length compatible with whatever output device you are using.

Below is the format of a line with length = 60, and margin of 10.

MARGIN	
10 SPACES	PRINTED LINE LENGTH = 50.

TOTAL LINE LENGTH = 60.

13. INSTRUCTION

DO YOU WANT ALL FILES TO BE PROCESSED? Y/N  
?--

NOTE:

If the ALL option is taken, the files will be processed in the order they appear when a LIST DC F command is executed.

14. INSTRUCTION

IF THE ALL OPTION IS TAKEN, GO TO STEP 18.

15. INSTRUCTION

ENTER THE NUMBER OF PROGRAM FILES TO BE PROCESSED.  
?--/



16. INSTRUCTION

ENTER THE NAME OF FILE 1

ERROR MESSAGE IF FILE NAME IS NOT ACTIVE ON DISK  
"PROGRAM ----- NOT AN ACTIVE PROGRAM FILE".

17. INSTRUCTION

REPEAT STEP 16 UNTIL ALL FILE NAMES HAVE BEEN ENTERED. AFTER THE LAST  
FILE NAME HAS BEEN ENTERED, THE FOLLOWING MESSAGE IS DISPLAYED:

KEY RETURN(EXEC) TO RESUME  
?-

18. INSTRUCTION

ARE THE PARAMETERS OK? (Y/N)  
?-

19. INSTRUCTION

IF "N" IS ENTERED, GO TO STEP 1.

20. INSTRUCTION

GO to Processing Phase.

3.5 PROCESSING PHASE

1. MESSAGE

\*\*\*\* LOADING PROCESSOR \*\*\*\*

2. MESSAGE

INITIALIZING KEY WORD TABLE

3. INSTRUCTION

IF DISK INPUT, SKIP STEP 4.

4. INSTRUCTION

MOUNT INPUT TAPE IN UNIT #  
KEY RETURN(EXEC) TO RESUME

5. MESSAGE

PROGRAM EXECUTION CONTINUES UNTIL ALL FILES HAVE BEEN PROCESSED OR UNTIL  
THE PROGRAM INTERRUPT IS USED.

### 3.6 ERROR RECOVERY FOR PROCESSING MODULE

S.F. 0 - RESTART LISTING AT THE BEGINNING OF THE CURRENT FILE  
KEY "H" TO INTERRUPT EXECUTION AT THE END OF THE CURRENT FILE

### 3.7 ERROR RECOVERY FOR EXECUTION PHASE

THERE IS AN ERROR RECOVERY ROUTINE IN THE SECOND MODULE OF THIS PROGRAM. THE ERROR RECOVERY ROUTINE ALLOWS THE USER TO RESTART THE PROGRAM AT THE BEGINNING OF THE CURRENT FILE. SHOULD THE PRINTER RUN OUT OF PAPER THIS ROUTINE CAN BE USED.

THE INTERRUPT ROUTINE ALLOWS THE USER TO HALT THE EXECUTION OF THE PROGRAM AT THE END OF THE CURRENT FILE AND CONTINUE EXECUTION BY KEYING RETURN(EXEC). THIS ALLOWS THE USER TO STOP THE PROGRAM TO MOUNT ADDITIONAL PAPER IN THE PRINTER WITHOUT UPSETTING ANY OF THE INTERNAL COUNTERS AND POINTERS OR DISTURBING THE DISPLAY.

### 3.8 ERROR RECOVERY OPERATION

TO RESTART THE CURRENT FILE AT THE BEGINNING WITH THE SPECIFIED OPTION, KEY SPECIAL FUNCTION 0.

TO INTERRUPT PROGRAM EXECUTION, KEY THE CAPITAL LETTER H AND THE PROGRAM WILL STOP AT THE END OF THE CURRENT FILE.

## 4.0

### DISK AND TAPE DUMP

#### 4.1 INTRODUCTION

The dump utilities print the contents of files stored on tape or disk. From either type of source, three forms of dump can be obtained.

The first two forms print the hexadecimal and character equivalents of the binary contents of the file. They are called the Horizontal and Vertical dumps. Hexadecimal values below 20 cause a "." to be printed; values above 7F print @.

A third kind of dump, Data File Structure, prints the contents of a data file, field by field. It shows the type of field and the length.

The dumps can be output to a printer or CRT.

The following equipment is needed to operate this DUMP program: a minimum 8K System 2200B, a CRT unit, a keyboard, a cassette drive, a disk drive (if input is to be obtained from disk), the program tape, an output device, and this set of operating instructions.

The remainder of these instructions provide operating instructions and explanations of the procedure used with the DUMP program.

#### 4.2 STEP 1: INITIALIZATION

Place the program tape in the main tape drive (address code = 10A) and make sure it is rewound.

Key      CLEAR, RETURN(EXEC)  
          LOAD "T/D-DUMP", RETURN(EXEC)  
          RUN, RETURN(EXEC)

This procedure clears the memory of the System 2200B, loads the first module of the DUMP program, and begins execution of the first module.

A number of prompting messages appear on the CRT display throughout this DUMP program; these are to aid you when entering data required for execution of the DUMP program.

The first prompting message on the CRT display instructs you to remove the DUMP program tape from Unit 1 (main tape drive).

After touching RETURN(EXEC), the next information needed is the date. The date can be entered in any format; however, the maximum number of characters allowed is 8. The format MM/DD/YY always has 8 characters or less.

#### 4.3 STEP 2: SPECIFYING INPUT DEVICE

The next request is for the input device. Enter 1 if the programs or data files to be dumped are on a tape cassette; enter 2 if they are on a disk. Step 3A provides the procedure used when dumping programs from tape, and Step 3B provides the procedure used when dumping programs from a disk.

#### 4.4 STEP 3A: USING A TAPE CASSETTE AS INPUT DEVICE

If you choose to dump programs or data files from a tape cassette, the next CRT display requests you to enter the tape unit number. A table is shown on the CRT illustrating the legal address codes for tape drives.

Enter the number (1-6) corresponding to the desired tape drive; touch RETURN(EXEC).

The next CRT display concerns the number of files to skip before dumping a program or data file; the maximum entry here is 99.

After you have entered the number of files to skip, the dump program asks you to mount the tape containing the programs or data files to be dumped; touch RETURN(EXEC) when you are ready to proceed.

The name of the file to be dumped is requested; if the file on tape does not have a name, simply touch RETURN(EXEC).

Now, the desired format of output must be entered (i.e., hard copy or CRT display). Enter either 1 if you desire a hard copy printout of your programs or data files, or 2 if you desire CRT printout.

#### NOTE:

Choosing the CRT display as an output option is not recommended, as all three types of output described in Step 4 are designed for a minimum 80-column printing device (the CRT can display only 64 characters per line).

Step 4 provides a detailed illustration of the procedures involved in choosing the CRT display and a hard copy printer as output devices. Also discussed in Step 4 are the three different dump formats available with this DUMP program.

#### 4.5 STEP 3B: USING A DISK AS INPUT DEVICE

If you choose to dump programs from a disk, different prompting messages appear on the CRT than appear when using a tape cassette.

The first prompt concerns the disk drive which contains the programs and data files to be dumped:

1. Fixed
2. Removable

The words fixed and removable refer to disk drives on all four disk units: Models 2230, 2240, 2242, and 2243. The following table illustrates which drives are associated with fixed and which drives are associated with removable.

ADDRESS CODES	RELATED DISK DRIVES			
	Model 2230	Model 2240	Model 2242	Model 2243
(FIXED) 310, 320, 330, 340	Fixed	Left	Only Drive	Left
(REMOVABLE) B10, B20, B30, B40, B50, B60	Removable	Right	Illegal	Middle
(FIXED) 350, 360	Fixed	Left	Only Drive	Right

After specifying the disk drive, you are prompted to choose the disk address option. A table is shown on the CRT, indicating option numbers 1-6; these are associated with address codes 310-360 (if FIXED disk) or B10-B60 (if REMOVABLE disk).

Enter the option number corresponding to your chosen disk address.

The DUMP program continues by asking you to mount the disk containing programs or data files to be dumped in the specified disk drive; touch RETURN(EXEC) when you are ready to proceed.

Before choosing the form of output, you must enter the name of the file to be dumped.

**NOTE:**

A program or data file on disk must be cataloged if they are to be dumped.

Now, the option of output format appears (i.e., CRT display or hard copy). Enter either 1 if you desire a hard copy printout of your programs or data files, or 2 if you desire CRT printout.

NOTE:

Choosing the CRT display as an output option is not recommended, as all three types of output described in Step 4 are designed for a minimum 80 column printing device (the CRT can display only 64 characters per line).

Step 4 provides detailed operating instructions when using either the CRT display or a hard copy output device. Also included in Step 4 is a description of the three different DUMP formats available with this DUMP program.

#### 4.6 STEP 4: EXECUTION OF THE DUMP PROGRAM

Two forms of output are available with the DUMP program:

1. HARD COPY
2. CRT DISPLAY

As mentioned in Steps 3A and 3B, the CRT display is not recommended for use as an output option because of the size of the CRT screen.

If you do choose the CRT display as the output device, the next question to be answered concerns which physical records within a program or data files are to be dumped.

However, if you choose hard copy as the output device, additional information is required before specifying physical records to be dumped. This information includes entering a unit number for the output device (a table is shown on the CRT display indicating legal output device address codes), and readying the printer.

When entering the physical records within a file, a lettering format appears on the CRT:

DUMP PHYSICAL RECORDS WITHIN FILE (FFFF\*LLLL OR ALL)

A physical record consists of 256 bytes of memory; thus one file can contain a number of physical records. If you wish all the physical records in your program or data file to be dumped, enter ALL, RETURN(EXEC); if you wish only certain physical records to be dumped, enter the first physical record to be dumped\* (FFFF), an asterisk, and the last record to be dumped (LLLL), followed by RETURN(EXEC).

The next question to be answered concerns the type of dump:

- 1 - Horizontal
- 2 - Vertical
- 3 - Data File Structure

The Figures

Figures 1, 2, and 3 on the following pages illustrate the three different types of dumps.

\*Physical records are numbered, starting with the first record as 0, not 1.



BYTES	RECORD - 0	RECORD - 1	RECORD - 2	RECORD - 3
0 - 31	4044551D 50FFFFFF	50555449 40495459	52243D52 2428542C	00FF0410 9AFF1470
32 - 63	30000000 FF507044	36342050 24383A9C	54303261 FF02603A	5445322 20225245
64 - 95	103A90FF 48300D00	00FF0000 00FF4690	2981FF02 903H5224	483031B2 36343A9H
96 - 127	0D0000FD 20202020	20202020 20FFFFFF	52243D52 3430290D	FF05307A 52243D22
128 - 159	FFFFFFFF FFFFFFFF	FFFFFFFF FFFFFFFF	703A9AFF 109AFF14	24202200 0000FD05
160 - 191	FFFFFFFF FFFFFFFF	FFFFFFFF FFFFFFFF	45444F52 22202252	492H1674 2028192B
192 - 223	FFFFFFFF FFFFFFFF	FFFFFFFF FFFFFFFF	9AFF0820 3H52243D	203H1672 2429300C
224 - 255	FFFFFFFF FFFFFFFF	FFFFFFFF FFFFFFFF	02603A9F 02603A9F	02194679 81FF0650
BYTES	RECORD - 0	RECORD - 1	RECORD - 2	RECORD - 3
0 - 31	00FF0410 A244554D	50555449 40495459	52243D52 2428542C	00FF0410 9AFF1470
32 - 63	322C5224 312C5124	36342050 24383A9C	54303261 FF02603A	5445322 20225245
64 - 95	23603H4D 304D2B31	303A9AFF 22103A90	2981FF02 903H5224	483031B2 36343A9H
96 - 127	44202022 3B4F2D31	3H9E493D 3H9E493D	52243D52 3430290D	FF05307A 52243D22
128 - 159	322C2849 2B31292H	33322D31 3B000000	703A9AFF 109AFF14	24202200 0000FD05
160 - 191	20000000 FF01709E	4H3D31B2 33323A4B	45444F52 22202252	492H1674 2028192B
192 - 223	02203H4B 3D313A54	3D320000 00FD0220	9AFF0820 3H52243D	203H1672 2429300C
224 - 255	292F343C 3EC5284B	2031292F 342981FF	02603A9F 02603A9F	02194679 81FF0650

Figure 4.1



DATAFILE

9/3/74

PAGE 1

PHYSICAL RECORD - 5

LOGICAL RECORD - 1

PHYSICAL RECORD WITHIN LOGICAL RECORD - 1

FIELD	TYPE	LENGTH
1	ALPHA	2
2	ALPHA	2
3	ALPHA	5
4	ALPHA	5
5	ALPHA	8
6	ALPHA	8
7	ALPHA	8
8	ALPHA	8
9	NUMERIC	8
10	NUMERIC	8
11	NUMERIC	8
12	NUMERIC	8
13	ALPHA	8
14	ALPHA	8
15	ALPHA	1
16	ALPHA	1
17	ALPHA	8
18	ALPHA	8
19	NUMERIC	8
20	NUMERIC	8
21	ALPHA	21
22	ALPHA	21

BYTES USED - 192

LOGICAL RECORD LENGTH - 192

Figure 4.3

The first type of dump, HORIZONTAL, must be printed on a Model 2221 or 2261 Printer, or the Model 2201 Output Writer. The format of the horizontal dump requires more columns than are available on the Model 2231 or 2241 Printers, or the Model 2202 Plotting Output Writer.

The horizontal printout (see Figure 4.1) provides the bytes of each physical record in 32 byte increments, the hex codes of all characters in the physical record and, in the right column, the characters of the program or data file.

NOTE:

Any hex code < 20 is printed as a period (.) in the character column (right column); any hex code > 80 is printed as @.

The second type of dump, VERTICAL, can be printed on any output device except the CRT display, as the dump requires more than 64 columns.

The vertical printout (see Figure 4.2) provides the bytes of each physical record in 64 byte increments, the characters of the program or data file, and directly below each character, its corresponding hex code.

The third type of dump, DATA FILE STRUCTURE, can be printed on any output device, including the CRT display. However, only data files can be dumped using this type of dump; no program can be dumped using this format.

The data file structure printout (see Figure 4.3) provides the number of physical records, the number of logical records, the number of physical records within a logical record, the fields in each physical record, the type of data in each field (i.e., alpha or numeric), the length of each field (in bytes), the total number of bytes used in a physical record, and the total number of bytes used in a logical record.

After you have specified the type of dump you desire, the CRT displays the option to accept (enter 1) or reject (enter 9) the entered parameters.

If you reject the parameters, the dump program returns you to the date; from this point, you can reenter all parameters correctly.

If you accept the parameters, the System 2200B begins dumping your programs or data files in the specified format.

NOTE:

If, at any time during the dump process, you wish to halt the program execution, enter H. This stops program execution at the end of the physical record being processed, without changing the data in the memory of the System 2200B. If you enter H, the PROCESSING INTERRUPT appears on the CRT display (see discussion below).

After all program or data files have been dumped, the CRT displays END OF FILE if you specified ALL files to be processed, or UPPER LIMIT REACHED if you specified a specific range of files to be processed (i.e., FFFF\*LLLL).

Directly below this message, the following display appears:

PROCESSING INTERRUPT KEY 1 - CONTINUE 2 - MODIFY 3 - REUSE 4 - END

You can key 1 - CONTINUE only if you have halted program execution by keying H.

Keying 2 - MODIFY returns you to the following message (see Step 4):

DUMP PHYSICAL RECORDS WITHIN FILE (FFFF\*LLLL OR ALL)

Keying 3 - REUSE returns you to the first request, the date (see Step 1).

Keying 4 - END signifies that all programs or data files have been dumped; the END PROGRAM message appears on the CRT, showing you the exact amount of memory remaining in your System 2200B.

## 5.0

### DISK SORT

#### 5.1 INTRODUCTION

##### 5.1.1 TERMINOLOGY

The following terminology is employed throughout this chapter:

- o Input file - The cataloged data file on disk which is to be sorted.
- o Input records - The records in the input file which are to be sorted (i.e., which are not yet sorted).
- o Output file - After the sort is complete, the cataloged data file on disk containing records from the input file, now sorted in ascending order.
- o Output records - The sorted records in the output file.
- o Field - One or more bytes in a record on disk containing a single value (alpha or numeric). Individual fields in a record are distinguished by a start-of-value (SOV) control byte prefixed to each field when information is stored on the disk. Each field is created by a single argument in the DATASAVE DC argument list when the record is written.
- o Key value - The value(s) stored in the designated key field(s) in each record. If only one key field is specified, the key value is the value stored in the specified key field in each record. If more than one key field is specified, the values are extracted from all designated key fields, and concatenated in the specified order to form a single extended key value. The key values are then sorted in ascending order as the first phase of the Disk Sort.
- o String - A sub-file of records or keys sorted in ascending order.

## 5.1.2 GENERAL SYSTEM OVERVIEW

The Disk Sort Utility is designed to sort a cataloged data file stored on any Wang 2200 series disk. The sort program reads records from a designated data file on disk (referred to as the "input file"), and writes them, sorted in ascending order, into a second cataloged file (referred to as the "output file"). The input records may be sorted on one or more key fields (up to a maximum of five).

A key sort is employed by the program to sort the input file. In a key sort, the key values are first extracted from the input records and sorted; once all keys have been sorted, the input records associated with the keys are directly accessed and written in the output file.

The actual sorting procedure is divided into three phases, or "passes": an internal sort phase, a merge phase, and an output phase. In the first phase of the sort, input records are read sequentially from the input file; as each record is read into memory, the program extracts its key (or keys). The programmer has the option of specifying several key fields (up to a maximum of five), to be combined in any order to form the sort key for each record. If more than one key field is specified, the designated keys are extracted in the order designated by the programmer, and concatenated into a single sort key for that record. Each sort key is stored, along with the absolute sector address of the input record from which it was extracted (and a record number, for blocked records) in a sort key buffer (one buffer for each sort key). The number of sort key buffers in memory (and hence, the maximum number of sort keys which can be held in memory at one time) is dependent upon the size of the sort keys (they may range from one to 64 bytes in length) and the size of available memory. However, the number of sort key buffers which can be set up by the program cannot exceed 63, even in configurations where additional memory space is available. The total number of records which can be sorted by the disk sort is 65,534.

When the sort key buffers are filled with keys and pointers, the keys are internally sorted using the replacement/selection method. Lists or "strings" of keys are sorted in ascending order and written into a temporary work file, located in the non-cataloged area of a specified disk platter. If all input keys can be stored in memory at once, only one sorted string is written; the program then skips phase two (merge pass), and proceeds directly to phase three (output).

In the normal case, the input file will contain more keys than can be held in memory at one time. Thus, two or more sorted strings are created in Pass 1 (the internal sort), and stored in the work file on disk. It now becomes necessary to merge the several strings into one sorted string of keys and pointers. The program proceeds automatically to Pass 2, the merge pass, in which the strings are read from the work file and merged into a successively larger sorted string. At the completion of the merge phase, a single extended sorted string is stored in the work file on disk.

With the completion of the merge operation, the program passes to the output phase, Pass 3. In the output phase, the keys (now sorted in ascending order) are sequentially read back from the work file. As each key is read, the pointer (absolute sector address) associated with it is extracted and used to directly access the sector in which the input record containing the key(s) is stored. The input record is then written into the output file. Thus, records are read from the input file and written to the output file in the order in which their associated keys are listed in the temporary work file. Since the keys are sorted in ascending order, the output records are written in ascending order. At the completion of the output phase, the output file contains records from the input file, sorted in ascending order. The input file itself remains unaltered by the sorting procedure.

### 5.1.3 THE DISK SORT SYSTEM

The Disk Sort Utility is broken down into a number of program overlays, each of which performs a specific set of functions, and each of which is called in automatically as the need arises. The system contains the following overlays or modules.

<u>Name</u>	<u>Function</u>
DISKSORT	Enters input specifications
DSS200BA	Generates code
DSS201AA	Pass 1 - Internal sort
DSS202AA	Pass 2 - Merge
DSS203AA	Pass 3 - Writes output file

### 5.1.4 RESTRICTIONS AND CONVENTIONS

The Disk Sort Utility imposes certain restrictions on the input file. In order to be sorted, the input file must conform to the following conventions:

1. The input file must be cataloged (i.e., it must have been opened initially with a DATASAVE DC OPEN statement); it must contain an end-of-file trailer record (written with a DATASAVE DC END statement) so that the number of sectors used is entered in the Catalog Index.
2. The number of records to be sorted may not exceed 65,534.
3. Input records may not exceed one sector in length.
4. All input records must have an identical format (this means that the order, length, type and number of fields in each record must be the same for all records).
5. Input records may be blocked (i.e., more than one record per sector), but the block length cannot exceed one sector.



6. Blocked records must be written in array form, with individual fields from each record grouped in a single array. For example:

```
DIM A$(4)3, B(4), C$(4) 20
DATASAVE DC #1, A$(), B(), C$()
```

Here four records are written in a sector, each record consisting of an A\$(X), B(X), AND C\$(X). However, the fields for each record are not stored sequentially. Instead, the first field in each of the four records is stored in A\$(), the second field in each record in B(), and the third field in C\$().

7. If the input records are blocked, each sector must contain the same number of records (all records must have the same format). The last sector of the file may contain one or more "dummy" records; in this case, the key fields of the dummy records should be assigned key values higher than all other keys in the file. For alpha keys, HEX(FF) is the highest possible value (e.g., A\$ = HEX(FF)), or for numerics, 9 E99 is the highest value. Non-key fields in the dummy records may be filled with random data. The format of dummy records must, of course, be consistent with the format of all other records in the file.
8. The number of fields in each record may not exceed 40.
9. The number of blocked records in each sector may not exceed 40.
10. If records are blocked, the total number of fields in each sector may not exceed 63.
11. The user may specify a maximum of five key fields in a record, in any order. The program then concatenates the specified fields in the order indicated to form a single extended key value for each record. The total length of the key value for each record may not exceed 64 bytes (not including SOV bytes). (Note: Each numeric field occupies eight bytes, not counting the SOV byte. Each alphanumeric field occupies either (a) the number of characters in the literal string, if a literal string was specified directly in the DATASAVE DC argument list, or (b) the dimensioned size of the alpha variable or array element containing the key value.)
12. An entire file, or any part of a file, may be sorted. If only part of a file is to be sorted, the ordinal number of the first record to be sorted (relative to the first record in the file), as well as the total number of records to be sorted, must be specified.

13. The program establishes a temporary work file in the non-cataloged area (outside the Catalog Area) on the user-specified platter. Normally, the size of the temporary work file is less than that of the input file. In all cases, a MINIMUM of 21 sectors are required for the temporary work file. Prior to initiating the sort procedures, the program automatically calculates the number of sectors required for the temporary work file, and returns an error message if there is insufficient space in the non-catalog area. (Note: the size of the Catalog Area and, hence, of the non-catalog area, is established initially with the SCRATCH DISK statement, wherein the 'END' parameter specifies the last sector to be reserved for the Catalog Area. The size of the Catalog Area can be altered with a MOVE END statement, if it is not completely filled with cataloged files.)
14. The output file must be a cataloged file. It may be a previously cataloged file, or a new file, opened by the Disk Sort program. The input file cannot be used as the output file; two distinct files are required. For blocked files, the output file contains the same number of records per sector as the input file.
15. The platter containing the input file and the platter containing the output file must be mounted at the start of the program, and must remain mounted throughout the program.

#### 5.1.5 THE 'EXEC' INSTRUCTION

The first module of the Disk Sort Utility (program DISKSORT) requests the operator to enter input specifications for the sorting operation. After each requested item is keyed in, the operator must touch the EXECUTE (CR/LF) key on the Model 2215 BASIC Keyword Keyboard, or the RETURN(EXEC) key on the Model 2222 Alphanumeric Typewriter Keyboard. Throughout the operating instructions which follow, a single term, "EXEC", is used to denote both of these keys.

#### 5.1.6 MACHINE REQUIREMENTS

The minimum system configuration required for the Disk Sort Utility is:

- One System 2200B with 8K of memory;
- One tape cassette drive;
- One disk drive (Model 2230, 2240, 2242, or 2243).

Because the Disk Sort is designed to run on all Wang 2200 series disk drives, and also because the speed and efficiency of the sorting operation increase with the amount of available memory, the program is designed so that the user may specify the memory size and disk platter size for his own configuration.

The variable M is used to store the size of the System 2200B memory, while the variable D holds the disk platter size. These variables are initially set to the following values:

M = 8 (8K system)  
D = 9792 (Model 2230-3)

If M = 8, the program will run on a larger machine, but it cannot take advantage of the additional memory. If D = 9792, the program will run on a smaller disk drive, but it cannot accurately determine if there is sufficient temporary work file space in the non-catalog area.

In systems with a memory configuration larger than 8K, variable M should be altered to reflect the size of available memory. M can be assigned a new value by rewriting line 480 in the program. Line 480 normally reads:

480 M = 8

For a system with a 12K memory, line 480 should be rewritten:

480 M = 12

For a 16K system:

480 M = 16

Since the maximum System 2200B memory configuration is 32K, the maximum value which M can assume is 32.

In systems with a disk drive smaller than the Model 2230-3, the value of D should be altered to reflect the number of sectors available on a disk platter in the available disk drive. D can be assigned a new value by rewriting line 490 in the program. Line 490 normally reads:

490 D = 9792

For a Model 2230-2, which has 4800 sectors on each platter, line 490 should be rewritten:

490 D = 4800

For a Model 2230-1, which has 2400 sectors per platter:

490 D = 2400

The number of sectors per platter for each 2200 series disk drive is listed below.

Model 2240-1	-	512
Model 2240-2	-	1024
Model 2242	-	1024
Model 2243	-	1024
Model 2230-1	-	2400
Model 2230-2	-	4800
Model 2230-3	-	9792

NOTE:

The changes to lines 480 and 490 must be made in program DISKSORT (the first module). The changes should be made after keying LOAD and prior to keying RUN.

## 5.2 OPERATING INSTRUCTIONS

Instruction	Explanation	Possible Error Message
1. Mount the program tape in the console tape unit.		
2. Key: RESET		
3. Enter: CLEAR Key: EXEC		
4. Enter: LOAD "DISKSORT" Key: EXEC		
5. The system loads the first module of the program.	If lines 480 and 490 are to be rewritten, the change should be made at this point, after the first module has been loaded, but before it has been run.	
6. Enter: RUN Key: EXEC		
7. The system displays: DISK SORT - ENTER SPECIFICATIONS		
8. The system displays: INPUT FILE NAME?	The input file must be a cataloged disk file.	
Enter the name of the input file. Key: EXEC		
9. The system displays: INPUT PLATTER (FOR R)?  Enter: F, R, or 3 Key: EXEC	F = Fixed Platter (Model 2230) = Left Platter (Model 2240) = Single Platter (Model 2242) = Platter #1 (Model 2243) R = Removable Platter (Model 2230) = Right Platter (Model 2240) = Platter #2 (Model 2243) 3 = Platter #3 (Model 2243)	2,3,4

10. The system displays: STARTING RECORD # TO BE SORTED (1 IF ALL)?
- To sort the entire file, enter 1.  
To sort part of the file, enter the number of the first record to be sorted.  
Key: EXEC
- The Disk Sort will sort a maximum of 65,534 records. For larger files, it is necessary to sort parts of the file, and subsequently merge the parts. This additional merge must be carried out by user-supplied software.
- 5,8
11. The system displays: NUMBER OF RECORDS TO BE SORTED (OR ALL)?
- To sort the entire file, or all the remaining records starting with the record specified in Step 10 above, enter ALL.  
To sort a specific number of records, enter the number of records to be sorted.  
Key: EXEC.
- If the number of records specified is greater than the number of records in the file (starting with the record specified in Step 10 above), then all remaining records are sorted.
12. The system displays: RECORDS PER BLOCK?
- Enter the number of records per sector (1 if not blocked).  
Key: EXEC
- 7,8
13. The program checks the data entered in steps 10-12 above against the length of the file and against a sample record.
- 9,10,11,  
12,13,14

14. The system displays:  
NUMBER OF KEY FIELDS  
(1 TO 5)?
- Enter the number of fields to be included in the sort key.  
Key: EXEC
- For example, the record contains: 8,15  
A\$ = account  
N\$ = name  
S\$ = address  
Z\$ = zip code
- To sort by zip code and name, enter 2 for the number of key fields, enter 4 for sequence number of key field 1, and 2 for sequence number of key field 2.
15. For each key field, the system displays:  
ENTER SEQUENCE NUMBER OF KEY FIELD x  
IN RECORD
- Enter the number of the field within the record which is to be the 1st, 2nd, etc., sort key field (see the explanation for step 14).
16. The program calculates sort blocking and sort buffer space, based on the length of the sort key.
17. The system displays:  
WORK FILE PLATTER  
(F OR R)?
- F = Fixed Platter  
(Model 2230)  
Left Platter  
(Model 2240)  
Single Platter  
(Model 2242)  
Platter #1  
(Model 2243)
- R = Removable Platter  
(Model 2230)  
Right Platter  
(Model 2240)  
Platter #2  
(Model 2243)
- 3 = Platter #3  
(Model 2243)
- The work file occupies the uncataloged (TEMP) area of any platter. Space required, in sectors, is approximately:  
$$25 + L(S + 5)/R$$
where L = length of file to be sorted, in sectors,  
S = sort key length, and  
R = record length.
- 2,19
- Enter: F, R, or 3  
Key: EXEC

18. The system displays:  
OUTPUT FILE NAME?  
  
Enter the name of  
the output file  
(sorted file).  
Key: EXEC
- The output file may be cataloged  
already. If not, it will be  
opened by the program.
19. The system displays:  
OUTPUT PLATTER  
(F OR R)?  
  
Enter: F, R, or 3  
Key: EXEC
- 2,20
20. The system displays:  
IS OUTPUT FILE  
CATALOGED? (Y OR N)?  
  
If the output file  
has been previously  
cataloged, enter Y  
and key EXEC, or  
key EXEC only.  
  
If the output file  
is not already cat-  
aloged, enter N.  
Key: EXEC
- The sort program will reserve  
the exact amount of space re-  
quired for the output file.  
If more space is desired, the  
output file must be cataloged  
in advance with a DATASAVE DC  
OPEN statement.
- 21,22,33,  
24
21. The program now runs  
without further op-  
erator intervention.  
The sequence of events  
is as follows:
22. The system loads the  
2nd module of the  
program. The second  
module generates code  
for the following  
steps, based on the  
input specifications.
- 1,24



- |  |  |                        |
|--|--|------------------------|
| <p>23. The system loads the third module of the program. The system displays:<br/>PASS 1 - INTERNAL SORT<br/>The program then proceeds with the internal sort.</p> | <p>A number of input records are read, and their keys extracted and stored. The number of keys which can be stored and sorted at once is dependent upon the length of the keys and the size of available memory. (Available memory space is calculated on the basis of the value of M, assigned at line 480 of module 1.) However, in all cases a maximum of 63 keys can be stored and sorted.</p> | <p>1,24,25,<br/>26</p> |
| <p>24. The system loads the fourth module of the program. The system displays:<br/>PASS 2 - MERGE<br/>The program then proceeds with the merge pass.</p>           | <p>The sorted strings are read from the temporary work file, and merged into a single extended sorted string, which is written back into the work file.<br/>Note: If all records can be held in memory at once, only one sorted string is created, and the merge pass is obviated.</p>   |                        |
| <p>25. The system loads the fifth module of the program. The system displays:<br/>PASS 3 - OUTPUT<br/>The program then proceeds with the output pass.</p>          | <p>Records are read from the input file and written to the output file in the order in which their keys appear in the sorted string.</p>   | <p>1,24,28</p>         |
| <p>26. The system displays:<br/>INPUT RECORDS XXXXX<br/>OUTPUT RECORDS XXXXX<br/>STOP END OF SORT</p>  | <p>The program is finished.</p>  |                        |

### 5.3 ERROR MESSAGES

Error Message	Explanation	Recovery
1. ERR 34	Tape read error.	Rerun the program with a new program tape.
2. RE-ENTER	Platter designated is not F, R, or 3.	Repeat the step. Enter F, R, or 3.
3. ERR 80	Input file not in catalog.	Locate input file with LIST DCF or LIST DCR, and check for the input file name. Rerun the program from step 6 (enter RUN, etc.).
4. INPUT INVALID	Input file is (a) not a data file, or (b) contains no records, or (c) has more than one sector per record or block of records, or (d) has one or more records whose format does not correspond to the format of the first record in the file.	Repeat step 8 with correct input file name, or stop.
5. Repeat prompt: STARTING RECORD # TO BE SORTED (1 IF ALL)?	Starting record number was less than 1 or not an integer.	Repeat step 10 with correct value.
6. Repeat prompt: NUMBER OF RECORDS TO BE SORTED (OR ALL)?	Number of records entered was less than 1, or not an integer, numeric, or ALL.	Repeat Step 11 with correct value.
7. Repeat prompt: RECORDS PER BLOCK?	Records per block entered was less than 1, greater than 40, or not an integer.	Repeat Step 12 with correct value.
8. ERR 29	A nonnumeric quantity was entered when a numeric quantity was requested.	Run the program from Step 6 (enter RUN, etc.) and enter a numeric value at the appropriate step.

- |  |   |  |
|--|---|--|
| 9. STARTING RECORD TOO HIGH                            | Starting record # entered is greater than the number of records in the file.  | Repeat Step 10 with correct starting record number.  |
| 10. TOO FEW TO SORT                                    | Less than one record to sort.   | Repeat from Step 10 (starting record number).  |
| 11. TOO MANY TO SORT                                   | More than 65,534 records to sort.   | Repeat from Step 10 (starting record number). If the file is longer than 65,534 records, it must be sorted in sections and merged later (with a user-supplied merge routine).                          |
| 12. STOP TOO MANY FIELDS                               | More than 40 fields per record or more than 63 fields per sector.   | The program will not sort this file.   |
| 13. STOP INVALID RECORD FORMAT                         | The sample record being examined is not in the proper format for a data record.   | The program will not sort this file.   |
| 14. STOP NOT BLOCKED AS SPECIFIED                      | The sample record, or block of records, being examined, does not have the same number of records per block as specified in Step 12, or the blocking is not in array form, i.e.,<br>DIM A\$(4)3, B(4), C\$(4)20<br>DATASAVE DC #n, A\$(), B(), C\$()<br>indicating 4 records per block, each containing an A\$(X), B(X), and C\$(X). | If records per block were entered incorrectly in Step 12, rerun the program, starting with Step 6 (enter RUN, etc.).<br><br>If the blocking is not in array form, the program will not sort this file. |
| 15. Repeat prompt:<br>NUMBER OF KEY<br>FIELDS (1 TO 5) | Number of key fields entered is less than 1, greater than 5, or not an integer.   | Repeat Step 14 with correct value.   |

- |   |   |   |
|---|---|---|
| 16. REENTER                                   | Sequence number of key field entered is less than 1, greater than the number of fields in the record, or not an integer.    | Reenter sequence number of key field.   |
| 17. STOP KEY LENGTH > 64                      | The total length of the sort key fields specified exceeds 64 bytes.   | The program cannot operate with a sort key longer than 64 bytes.  |
| 18. STOP NO BUFFER SPACE                      | Program error, machine error, or memory size less than 8K specified.  | Correct the error. Rerun the program.   |
| 19. WORK SPACE TOO SMALL                      | The uncataloged (TEMP) area of the specified platter is too small to accommodate the work files.                            | Repeat Step 17 with a different catalog area designation. The size of the Catalog Area can be altered with a MOVE END statement, if it is not filled with cataloged files. Otherwise, it may be necessary to mount another platter with a larger uncataloged (TEMP) area. |
|   | A program designed to run on the floppy disk (D = 1024) is being run on the 2230-3 disk (D = 9792).                         | Key: HALT/STEP<br>Enter: PRINT D<br>Key: EXEC<br>If D = 1024 and the program is being run on the 2230-3 disk:<br>Enter 490 D = 9792<br>Key: EXEC<br>Rerun the program, starting at Step 6 (enter RUN, etc.).  |
| 20. INPUT AND OUTPUT MAY NOT BE THE SAME FILE | The same file, on the same platter, is specified as both input and output. This would cause the input file to be destroyed. | Repeat Step 18 with a different output file name, or else designate a different platter for the output in Step 19.  |
| 21. REENTER                                   | An invalid character was entered in response to a yes (Y) or no (N) question.   | Reenter Y or N.   |

22. OUTPUT SPACE TOO SMALL	The output file designated does not have enough space to hold the output from this sort.	Repeat Step 18 with a different output file name (may be a new file).
23. ERR 80	Output file not found.	Rerun from Step 6 (key RUN, etc.). Correct output file name or designate it as "not cataloged".
24. ERR 72 and other machine error messages	See manual for explanation.	Rerun the program from Step 1.
25. STOP B	Program error, Pass 1	Rerun with another program tape.
26. STOP GEN	Program error, Pass 1	Rerun with another program tape.
27. STOP P	Program error, Pass 2	Rerun with another program tape.
28. ERR 43	Program error, Pass 3	Rerun with another program tape.

## 6.0

### TAPE SORT

#### 6.1 DEFINITION "SORT"

To rearrange a set of data items into a new sequence, governed by specific rules of precedence, and by extension, the program is designed to perform the above activity.

A tape sort begins by distributing the data into strings on work tapes (first pass). It then merges the tapes onto new work tapes, forming fewer, longer partial sequences (merge phase). The final merge (last pass) creates a single and sorted output sequence.

#### 6.2 GLOSSARY

##### Terminology

- o Input file - The data file which is to be sorted.
- o Input records - The records in the input file which are to be sorted (i.e., which are not yet sorted).
- o Field - One or more bytes in a record on tape containing a single value (alpha or numeric). Individual fields in a record are distinguished by a start-of-value (SOV) control byte prefixed to each field when information is stored on the disk. Each field is created by a single argument in the DATASAVE DC argument list when the record is written.
- o Key Value - The value(s) stored in the designated key field(s) in each record. If only one key field is specified, the key value is the value stored in the specified key field in each record. If more than one key field is specified, the values are extracted from all designated key fields, and arranged in the specified order to form a single extended key value. The key values are then sorted in ascending order as the first phase of the Tape Sort.
- o String - A set of consecutive, adjacent items of similar type. Normally a bit string or a character string. In general, string means a sequence of variable or arbitrary length, composed of bits, digits, or characters. The beginning and end, or beginning and length of the string, are usually made known to the computer. Characters arriving from a remote console, statements input to a compiler, messages of various kinds, and records on tape are strings.

In the context of tape sort, a string is a group of consecutive records that have been sorted into ascending order (a sub-file which becomes the total file when the sort is finished).

String-manipulating operations include concatenation (joining two strings to form one new longer one), comparison (testing for equality), scanning (finding the first of a specified character, all of a specified character, or the number of appearances of specified characters).

### 6.3 GENERAL

The 2200 Tape Sort Utility is made up of 7 modules, which are divided into three passes, PASS 0, 1, and 2.

Pass 0 accepts sort specifications from the operator and modifies the general sort programs for a particular case. PASS 1 sorts the records internally, using the replacement selection technique and writing sequences as "strings" of sorted records on the work tape. PASS 2 merges these sorted strings into one using the polyphase method.

There are 7 modules; line number 480 in the first module should be modified for the user's installation if greater than 8K memory (M = 12 for 12K memory up to M = 32 for 32K memory).

This is a one-time operation and once this is changed the entire 7 modules should be copied to another tape. The name of each module is the first statement in each module and must be preserved "SAVE NAME".

### 6.4 TAPE SORT

The Tape Sort contains the following 7 modules:

1. Enter input specifications
2. Miscellaneous set-up functions
3. PASS 2 - Merge
4. Subroutine - convert numeric sort keys back from sort format
5. Generate codes
6. PASS 1 - Internal Sort
7. Subroutine - convert numeric sort keys to sort format

The 2200 Tape Sort is designed to sort a data file stored on tape. It creates a file of output with records arranged in ascending order of one or more fields (up to 5) in the record.

### Three Main Sections

The Tape Sort is divided into three main sections. Pass 0 accepts input specifications from the operator and modifies the general sort program for the particular case. The program generates program statements to be inserted in the Pass 1 and Pass 2 programs.

Pass 1 is an internal sort. Input records are read and reformatted, with the sort key in one alpha field, and the other fields of the record grouped together, for more efficient processing. The replacement-selection method is used for the internal sort. As each output record is written, in sort sequence, an input record is read and reformatted to fill the buffer just vacated. The position of the input record in the sorted sequence is then determined, and the internal sort process continues until the last output record written to the sorted string is higher than any of the records in the sort buffers. Then the string is terminated and a new string begun, starting with the lowest sort key. Output strings are distributed to tapes by a horizontal distribution method. It assigns strings to tapes, whenever possible, so that there is a perfect distribution for a polyphase merge. If that is not possible, it maintains the number of dummy strings per tape as nearly equal as possible.

Pass 2 is a polyphase merge using from 3 to 6 tapes. Dummy strings are "merged" first to minimize the number of records processed.

On the last merge pass, the records are reformatted from the sort format to the original input format before they are written on the output tape. There is at least one merge pass to reformat the records, even if there is only one string to "merge".

Two input formats are accepted:

Format 0 (no labels):

No header record precedes the file. A hardware trailer record (DATASAVE END) follows the last block of data.

Format 1 (standard labels):

A header record of the following format precedes the data:

<u>Field</u>	<u>Length</u>	<u>Type</u>	<u>Contents</u>
1	3	Alpha	"HDR"
2	8	Alpha	File name
3	8	Numeric	Creation date
4	8	Numeric	Retention (days)
5	8	Numeric	Reel number

The rest of the block is unused.

The first field of each data record must be a record ID (alpha, length 2). End-of-file is determined by a warning record. The last physical tape block must contain a "3" in the second position of the first field (of the first record, if blocked). No other live information is contained in that tape block.



A trailer record of the following format follows the data:

<u>Field</u>	<u>Length</u>	<u>Type</u>	<u>Contents</u>
1	3	Alpha	"EOR" or "EOF"
2	8	Alpha	File name
3	8	Numeric	Creation date
4	8	Numeric	Retention (days)
5	8	Numeric	Reel number

Input Records May Be Blocked:

- 1) If input records are blocked, the unused spaces in the last tape block must contain padding records (high values in the sort key fields).
- 2) The record, or block of records, may not exceed one physical tape block (253 bytes of data, including start-of-value (SOV) bytes).
- 3) Input records must all have identical format (except for header and trailer records as described above).
- 4) Blocked records are assumed to have been written in array form:

```
DIM A$(4)3, B(4), C$(4)20  
DATASAVE A$(), B(), C$()
```

indicating 4 records per block, each containing an A\$, B, and C\$.

## 6.5 OPERATING PROCEDURES

Instruction	Explanation	Possible Error Message
1. Mount the program tape in the console tape unit.		
2. Key: RESET		
3. Enter: CLEAR Key: RETURN(EXEC)		
4. Enter: LOAD "TAPESORT" Key: RETURN(EXEC)		
5. The system loads the first module of the program.		1
6. Enter: RUN Key: RETURN(EXEC)		
7. The System displays: TAPE SORT - ENTER SPECIFICATIONS		
8. The system displays: TAPE FORMAT?		2,3
Enter 0 for unlabeled tape. Enter 1 for standard labels. Key: RETURN(EXEC)	If Format 0, proceed with Step 9, below. If Format 1, proceed with Step 10, below.	
9. (Format 0) The system displays: MOUNT INPUT TAPE, UNIT 10B KEY RETURN(EXEC) TO RESUME		
Mount the input tape on unit 10B. Key: RETURN(EXEC)		
Proceed with step 12, below.		

10. (Format 1) The system displays:  
INPUT FILE NAME?  
  
Enter the name of the input file.  
Key: RETURN(EXEC)
11. The system displays: 1,4  
MOUNT TAPE (name),  
UNIT 10B  
KEY RETURN(EXEC)  
TO RESUME  
  
Mount the designated input tape on unit 10B. To resume,  
Key: RETURN(EXEC)
12. The system displays:  
STARTING RECORD #  
TO BE SORTED  
(1 IF ALL)?  
  
Enter 1 to start sorting with the first record in the file; otherwise enter the number of the first record to be sorted.  
Key: RETURN(EXEC)
13. The system displays: 6  
NUMBER OF RECORDS TO BE SORTED (OR ALL)?  
  
Enter ALL to sort all the records from the starting number specified to the end of the file. Otherwise enter the number of records to be sorted.  
Key: RETURN(EXEC)
14. The system displays: 3,7  
RECORDS PER BLOCK?  
  
Enter the number of records per block.  
Key RETURN(EXEC)

15. The system reads the starting block on the input tape, determines the record format, and checks the blocking. 1,8,9,10,11
16. The system displays:  
NUMBER OF KEY FIELDS  
(1 TO 5)? 3,12
- Enter the number of fields to be included in the sort key.  
Key: RETURN(EXEC)
17. Per key field:  
The system displays:  
ENTER SEQUENCE  
NUMBER OF KEY FIELD  
X IN RECORD 3,13,14
- Enter the number of the field within the record which is to be the 1st, 2nd, etc., sort key field.  
Key: RETURN(EXEC)
18. The system displays:  
NUMBER OF CASSETTE  
DRIVES AVAILABLE  
FOR SORT? 3,15
- Enter 3,4,5, or 6  
Key: RETURN(EXEC)
- If Format 0 (no labels) proceed with step 22, below.
19. (Format 1) The system displays:  
OUTPUT FILE NAME?
- Enter the name of the output file to appear on the label.  
Key: RETURN(EXEC)

20. (Format 1) The system displays: 16  
TODAY'S DATE MM/DD/YY?
- Enter today's date. This date is converted to a  
Key: RETURN(EXEC) Julian date and becomes the  
creation date on the output  
tape and all scratch tapes.
21. (Format 1) The system displays: 3  
RETENTION PERIOD IN  
DAYS?
- Enter the retention  
period for the  
output tape.  
Key: RETURN(EXEC)
22. The system displays: 1,17,18  
MOUNT SCRATCH TAPE,  
UNIT 10C  
KEY RETURN(EXEC) TO  
RESUME
- Mount a scratch tape (Format 1) The program labels  
on the specified tape all scratch tapes "SORTWORK",  
unit (labeled scratch with today's date and a reten-  
tape if Format 1). tion cycle of -1.  
To resume:  
Key: RETURN(EXEC)
- If 4 tape sort, repeat  
step 22 with unit 10D.  
If 5 tape sort, repeat  
step 22 with unit 10E.  
If 6 tape sort, repeat  
step 22 with unit 10F.
23. The system copies the 1  
Pass 2 (merge) pro-  
gram to work tape 10C  
and generates code  
for Pass 1 and Pass  
2, based on the speci-  
fications entered.
24. The system loads the 1  
Pass 1 (internal  
sort) program and  
rewinds the program  
tape.

25. The system displays:  
 REMOVE PROGRAM TAPE  
 MOUNT SCRATCH TAPE  
 IN CONSOLE TAPE UNIT  
 KEY RETURN(EXEC)  
 TO RESUME
- Next operator intervention required is at the end of Pass 1, step 29, below. Allow 15-20 minutes for a full reel of input.
- 1,17,18
- Remove the program tape and mount a scratch tape (labeled scratch tape if Format 1) in unit 10A.  
 To resume,  
 Key: RETURN(EXEC)
26. The system displays:  
 Pass 1 - INTERNAL  
 SORT
27. The program performs the internal sort, reading the input tape and distributing strings of sorted records on the work tapes.
- 1,20,21,22,25
28. The system loads the Pass 2 (merge) program from unit 10C and rewinds the input tape, unit 10B.
- 1
29. The system displays:  
 REMOVE INPUT TAPE,  
 UNIT 10B  
 MOUNT SCRATCH TAPE,  
 UNIT 10B  
 KEY RETURN(EXEC)  
 TO RESUME
- 1,17,18
- Remove the input tape and mount a scratch tape (labeled scratch tape if Format 1) in unit 10B.  
 To resume,  
 Key: RETURN(EXEC)
30. The system displays:  
 PASS 2 - MERGE

31. The system merges  
the sorted strings  
into one output  
file.

1,23,24,25

32. The system displays:  
REMOVE AND LABEL  
OUTPUT, UNIT XXX

The sorted output is  
on the unit named  
above. Remove and  
label.

Note that the output tape is  
unpredictable and that this  
display message is the only  
way of knowing which reel the  
sorted output is on. It is  
important not to clear the  
screen until the output reel  
is removed and labeled.

33. The system displays:  
STOP END OF SORT

The program is finished.

## 6.6 ERROR MESSAGES

Error Message	Explanation	Recovery
1. ERR 34	Tape read error.	Rerun the program with a new tape.
2. Repeat prompt: TAPE FORMAT?	Tape format entered must be 0 or 1.	Repeat step 8.
3. ERR 29	A non-numeric quantity was entered when a numeric quantity was requested.	Reenter numeric quantity.
4. WRONG TAPE	The name on the tape label is not the same as the input file name entered, or the first record is not a standard header label.	Repeat from step 10.
5. Repeat prompt: STARTING RECORD # TO BE SORTED (1 IF ALL)?	Starting record number entered was less than 1 or not an integer.	Repeat step 12.
6. Repeat prompt: NUMBER OF RECORDS TO BE SORTED (OR ALL)?	Number of records entered was less than 1 or not an integer, numeric, or ALL.	Repeat step 13. Enter a number greater than 0, or ALL.
7. Repeat prompt: RECORDS PER BLOCK?	Records per block entered was less than 1, greater than 40 or not an integer.	Repeat step 14 with correct value.
8. STARTING RECORD TOO HIGH	End-of-file was reached on the input tape while searching for the starting record to be sorted.	Repeat from step 12.
9. STOP INVALID RECORD FORMAT	a) The record or block of records is longer than one physical tape block. b) The control bytes in the tape record are invalid. c) If Format 1, the first field in the record is not alpha, length 2.	The program will not sort this file.



- |   |   |  |
|---|---|--|
| 10. STOP TOO MANY FIELDS  | a) More than 40 fields per record.<br>b) More than 63 fields per block.   | The program will not sort this file.   |
| 11. STOP NOT BLOCKED AS SPECIFIED                                   | a) Records per block entered is not the same as actual records per block on the tape.<br>b) The blocking is not in array form, i.e.,<br>DIM A\$(4)3,B(4),C\$(4)20<br>DATASAVE A\$(),B(),C\$()<br>indicating 4 records per block, each containing an A\$,B, and C\$. | Rerun the program from step 6.<br><br>The program will not sort this file.   |
| 12. Repeat prompt:<br>NUMBER OF KEY FIELDS (1 to 5)?                | Number of key fields entered was less than 1, greater than 5, or not an integer.  | Repeat step 16.  |
| 13. RE-ENTER  | Sequence number of key field entered was less than 1, greater than the number of fields in the record, or not an integer.   | Reenter sequence number of key field.  |
| 14. STOP KEY LENGTH<br>64   | The total length of the sort key fields specified exceeds 64 bytes.   | The program will not handle a sort key longer than 64 bytes.   |
| 15. Repeat prompt:<br>NUMBER OF CASSETTE DRIVES AVAILABLE FOR SORT. | Number of cassette drives entered is not 3, 4, 5, or 6.   | Repeat step 18.  |
| 16. INVALID   | Date entered is not a valid date.   | Repeat step 20.  |
| 17. ERR 43  | Scratch tape does not have a header record in standard format (Format 1).   | Rewind the tape. Mount a labeled scratch tape. Enter: RUN XXXX, where XXXX is the statement number displayed.<br>Key: RETURN(EXEC) |
| 18. RETENTION CYCLE NOT EXPIRED                                     | The retention cycle has not expired on the tape mounted as a scratch tape.  | Mount another labeled scratch tape.<br>Key: RETURN(EXEC)   |

19. STOP BLOCKING	Program error.	Rerun with another program tape.
20. STOP B	Program error.	Rerun with another program tape.
21. STOP INPUT INVALID	The input file is not in the format specified.	The program will not sort this file.
22. STOP GEN	Program error.	Rerun with another program tape.
23. STOP O	Program error.	Rerun with another program tape.
24. STOP P	Program error.	Rerun with another program tape.
25. ERR 49	<p>End of physical reel of tape:</p> <p>a) When sorting a full reel of tape, the output tape must be at least as long as the input tape. Since the output tape is not known at the start, all scratch tapes must be long enough to hold the full output file.</p> <p>b) In unusual conditions the sort blocking can be so much less efficient than the input blocking that the capacity of a work tape can be exceeded.</p>	<p>Rerun the program.</p> <p>a) Cassette tapes are manufactured to fairly close tolerances. A 150-foot reel will hold <math>300 + 2</math> physical tape blocks. If the input reel never exceeds 295 blocks, there should be no problem.</p> <p>b) Sort half the file at a time, and merge the halves later.</p>

7.0  
TAPE/DISK & DISPLAY  
UTILITIES

7.1 INTRODUCTION

The System 2200 STANDARDS UTILITIES consist of a series of DEFFN' subroutines which can be appended to a program requiring the functions provided. The subroutines are listed in Table 7.1.

Table 7.1 STANDARDS UTILITIES Subroutines

Subroutine Number	Utility Name
'240	Open Output File
'241	End of Volume/File Output
'243	Accept Alpha Data (With Message)
'244	Accept Alpha Data (No Message)
'245	Accept Numeric Data (With Message)
'246	Accept Numeric Data (No Message)
'248	Position Cursor
'250	Open File Input
'251	Close Volume Input
'254	Operator Wait

To use STANDARDS UTILITIES, the programmer must include the following commoned variables in his program at a point where they will be executed prior to calling any of the subroutines: COM Q1, Q1\$8, W\$(3)8, Q4(3), Q5(3), Q5\$1, Q6\$64, W\$8, W3\$3, W4\$1.

The calling sequence for each subroutine specifies its internal variables. For further details see the System 2200 Reference Manual: GOSUB' and DEFFN'.

For use with the file-handling routines, a program must contain a SELECT statement for each Logical Drive (Disk Drive) in use. If a master file is to be read from Logical Drive No. 1, updated and written to Logical Drive No. 2, the two SELECT statements would be coded as follows:

SELECT #1 310

SELECT #2 B10

To access a third disk drive, SELECT #3 350 would be used. The following sections describe each utility subroutine in detail, with a listing of both source and compressed versions.

NOTE:

Any disks or tapes to be used for file storage with the STANDARDS UTILITIES '240 and '241 must be initialized with SCRATCH VOL. INITIALIZE. If they are not initialized the STANDARDS UTILITIES '240 and '241 yield execution errors.

## 7.2 OPEN OUTPUT FILE '240

GOSUB' 240 is used to obtain a scratch volume to be used for output. The calling sequence for GOSUB' 240 is:

```
GOSUB' 240 (W, W$(W), Q4(W), Q5(W))
```

where: W = a file number from 1 to 3,  
W\$(W) = file name of 8 characters,  
Q4(W) = retention cycle - number of days to retain this file before destruction, and  
Q5(W) = volume number which should be set equal to 1 for the first platter of this file.

When executed, GOSUB' 240 prints

"MOUNT SCRATCH VOL TO BECOME VOL## OF FILE##### - UNIT#" on line 0 of the CRT,

calls the Operator Wait utility '254,

reads the first record from the mounted volume and checks that the retention cycle has expired. (If the retention cycle has not expired, GOSUB' 240 instructs the operator to mount the proper volume. The check can be omitted by keying X in response to the Operator Wait utility.),

rewrites the first record of the mounted volume with a new creation date and retention cycle, and

erases lines 0, 1, 2 and 3 of the CRT.

In all the file manipulation subroutines, W is used to represent the logical unit. For each value of W, a SELECT statement must have been executed. When executing an OPEN, the "file name" is saved in W\$(W) and the volume number in Q5(W). For an output file, the retention cycle also is saved in Q4(W). Then when a CLOSE is processed (input or output), the values for the next volume are already known.

Q1 (today's Julian date) must be assigned by the user before calling for any output file processing.

Variables modified by '240 are: Q6, Q7, Q8, W0, Q6\$, W3\$, W\$, W1, Q4, Q5.

### 7.3 END OF VOLUME/FILE OUTPUT '241

GOSUB' 241 either closes a volume for a multi-volume file or closes a file. The calling sequence for GOSUB' 241 is:

```
GOSUB' 241 (W, W3$)
```

where: W = the number of the logical unit to which the file is assigned, and  
W3\$ = "EOV" for End-of-Volume and "EOF" for End-of-File.

#### End-of-Volume

If "EOV" is specified, GOSUB' 241 writes an End-of-Volume record on the correct disk and asks the operator to remove and label it as volume number ## of file "filename". The volume number is incremented by 1 and GOSUB' 240 (described earlier) is automatically executed. Lines 0, 1, 2 and 3 of the CRT are used and cleared by GOSUB' 240.

The variables modified when "EOV" is specified are the same as GOSUB' 240, Open Output File.

#### End-of-File

If "EOF" is specified, an EOF record is written. The volume number is not incremented and GOSUB' 240 is not executed. The variables modified are Q6, Q7, Q8, Q6\$ and W0.

### 7.4 ACCEPT ALPHA DATA (WITH MESSAGE) '243

GOSUB' 243 prints a prompt message on line 1 and a series of prompting dashes on line 2 equal to the maximum allowable field size to be entered by the operator. The information keyed in by the operator appears on line 2, replacing the prompting dashes. The calling sequence for GOSUB '243 is:

```
GOSUB' 243 (Q6$, Q0)
```

where: Q6\$ = prompt message, and  
Q0 = maximum number of characters to be entered.

The alpha input accepted by GOSUB' 243 is returned in the variable Q6\$. If the alpha input exceeds the value of Q0, the word "RE-ENTER" appears on line 3, and line 2 is replaced by the prompting dashes. Line 3 is automatically cleared upon exiting the routine.

#### NOTE:

If Q0 = 0, the field size check is omitted. Variables modified by the routine are: Q6\$, Q6, Q7, Q8.

## 7.5 ACCEPT ALPHA DATA (NO MESSAGE) '244

GOSUB' 244 is identical to '243, except that no prompt message is displayed on the CRT. It is used to ask for additional data from the operator without repeating the previous prompt message, or the prompt message is of a form that cannot be created by '243 (i.e., part of the message involves printing a stored value). The calling sequence for GOSUB '244 is:

```
GOSUB' 244 (Q0)
```

where: Q0 = maximum number of characters to be entered.

Alpha input accepted by GOSUB' 244 is returned in variable Q6\$. If the alpha input exceeds the value of Q0, the word "RE-ENTER" appears on line 3 and line 2 is replaced by the prompting dashes. Line 3 is automatically cleared upon exiting the routine.

### NOTE:

If Q0 = 0 the field size check is omitted. Variables modified by the routine are: Q6\$, Q6, Q7, Q8.

## 7.6 ACCEPT NUMERIC DATA (WITH MESSAGE) '245

GOSUB' 245 prints a prompt message on line 1 and a series of prompting dashes on line 2 indicating the maximum number of digits to be entered before and after the decimal point. The decimal point position is indicated by a slash (/).

The numeric data keyed in by the operator appear on line 2, replacing the prompting dashes. The calling sequence of GOSUB' 245 is:

```
GOSUB' 245 (Q6$, Q2, Q3)
```

where: Q6\$ = prompt message,  
Q2 = maximum number of digits before the decimal point, and  
Q3 = maximum number of digits after the decimal point.

The prompt message is output on line 1 of the display. Q2 dashes, followed by a slash, followed by Q3 dashes are output in line 2 of the display to prompt the field size. The numeric input returned by the routine is stored in variable Q9.

If the numeric input exceeds the image (Q2 and Q3), the word "RE-ENTER" appears on line 3, and line 2 is replaced with prompting dashes. Only a positive number is allowed if Q2 is greater than or equal to 0. If Q2 is less than zero, a positive or negative number is allowed.

Line 3 is automatically cleared upon exiting the routine. Variables modified by the routine are: Q6\$, Q6, Q7, Q8, Q9, W0.

## 7.7 ACCEPT NUMERIC DATA (NO MESSAGE) '246

GOSUB' 246 is identical to '245 except that no prompt message is displayed on the CRT. The calling sequence for GOSUB' 246 is:

```
GOSUB' 246 (Q2, Q3)
```

where: Q2 = maximum number of digits  
before the decimal point, and

Q3 = maximum number of digits  
after the decimal point.

Variables modified are: Q6\$, Q6, Q7, Q8, Q9, W0.

## 7.8 POSITION CURSOR '248

GOSUB' 248 positions the cursor at any point on the CRT and optionally erases all or part of the screen from that point on. The calling sequence for GOSUB' 248 is:

```
GOSUB' 248 (Q6, Q7, Q8)
```

where: Q6 = row,  
Q7 = column, and  
Q8 = number of lines to erase.

The screen is assumed to consist of 16 rows numbered 0 through 15, and 64 columns numbered 0 through 63. When executed, GOSUB' 248 positions the cursor at the indicated row and column and, if variable Q8 is equal to 0, characters on the screen are not erased. If Q8 is equal to 1, the rest of the row (line) on which the cursor is positioned is erased. If Q8 is greater than 1, an additional number of lines equal to the value of Q8-1 are erased from column Q7 to column 63. If the erase option is used, variable Q6\$ is set to spaces. The variable W0 is used as an index.

## 7.9 OPEN INPUT FILE '250

GOSUB' 250 makes a specified volume of an input file available for processing. The calling sequence for GOSUB' 250 is:

```
GOSUB' 250 (W, W$(W), Q5(W))
```

where: W = the number of the logical unit to which the file is  
assigned,  
W\$(W) = the name of the file, and  
Q5(W) = the number of the first volume to be read by the  
program (normally 1).

When executed, GOSUB' 250 asks the operator to mount the specified volume of the specified file on the requested logical unit. When the disk is mounted, the header record is read and the file name and volume number are checked. If the name and number do not check, the operator is asked to mount the correct volume. The check can be omitted by the operator keying X. When control is returned to the calling program, the cassette is positioned at the first data record of the file.

The variables modified are; Q6, Q7, Q8, W0, Q6\$, W3\$, W\$, W1, Q4 and Q5. Lines 0, 1, 2 and 3 of the CRT are used and cleared by GOSUB' 250.

#### 7.10 CLOSE INPUT VOLUME '251

GOSUB' 251 is used to terminate processing of the current volume of an input file and to make the next volume of the file (if any) available for processing. The calling sequence for GOSUB' 251 is:

```
GOSUB' 251 (W)
```

where: W = file number.

When executed, GOSUB' 251 assumes that the disk is positioned just past the last data record of the volume. The trailer record is read and checked to determine if it is End-of-Volume or End-of-File. If it is End-of-File, control is returned to the calling program with variable Q5\$ = "F". If End-of-Volume, the next volume is requested and automatically opened through the use of subroutine '250, and, when mounted and opened, control is returned with variable Q5\$ = "R".

In either case, the variables modified are Q6, Q7, Q8, W0, Q6\$, W3\$, W\$, W1, Q4 and Q5.

#### 7.11 OPERATOR WAIT '254

GOSUB' 254 is used whenever an outstanding request (normally displayed on line 0 of the CRT) must be entered by the operator before processing can continue. The calling sequence for GOSUB' 254 is:

```
GOSUB' 254
```

When executed, GOSUB' 254 displays the message "KEY RETURN(EXEC) TO RESUME" on line 1 of the CRT. When the operator responds, lines 1, 2 and 3 of the screen are cleared and control is returned to the calling program. If the operator enters any characters prior to the EXEC, a maximum of eight characters is returned in variable Q6\$.

Variables modified by this routine are Q6, Q7, Q8\$, W4\$, Q8 and W0.



## 7.12 VARIABLES

VARIABLE	DESCRIPTION
Q1\$8	Today's date in calendar form; e.g., 02/15/71
Q5\$1	"V" for End-of-Volume, "F" for End-of-File
Q6\$64	Message to operator, reply from operator
Q8\$1	"A" for alphanumeric data, "N" for numeric
W\$8	File name (input): 8 characters
W3\$3	"HDR", "EOV", or "EOF" header/trailer type
W4\$1	Scratch variable in '254

## 8.0

### SCRATCH VOLUME

### INITIALIZATION

#### 8.1 GENERAL

Scratch Volume Initialization initializes a tape or disk for use with the open and close subroutines ('240, '241) of the Standards Utilities. The system tape for Tape Utilities contains Scratch Volume Initialization for tape, while the Disk Utilities system tape contains the disk version.

Scratch Volume Initialization itself uses the Standards Utilities which must, therefore, be loaded prior to execution.

Any tape or disk to be used with subroutines '240 or '241 must first be initialized with this utility.

#### NOTE:

To stop the program, key END in response to the message "KEY RETURN(EXEC) TO RESUME".

## 9.0

### DATE ROUTINES

#### 9.1 INTRODUCTION

The STANDARDS DATE SUBROUTINES are a group of subroutines which facilitate the entry and manipulation of dates. They are accessed via the GOSUB (line number) statement. They utilize subroutines in the Standards Utilities which must be loaded, therefore, during date subroutine operations.

The Accept Date routine prompts the operator for entry of the date in calendar form. It then converts the entry to Julian form and displays both forms for operator verification. It is normally executed before the other date subroutines, and initializes certain variables upon which the others depend.

If any of the other subroutines are executed prior to Accept Date, U() and U\$ must be initialized as follows:

```
U(1)=0           U(5)=121           U(9) =244
U(2)=31          U(6)=152           U(10)=274
U(3)=60          U(7)=182           U(11)=305
U(4)=91          U(8)=213           U(12)=335
U$ = "0123456789"
```

#### 9.2 TWO DIGIT PACK

##### Operation:

Separates the last two digits of V4 and inserts into U3\$, beginning column V5.

##### Input:

U3\$ = original string,  
V4 = positive or zero number, to be packed, and  
V5 = positive integer; column of U3\$.

##### Output:

U3\$

##### Example:

If U3\$ = "XXXXXXXX" and this subroutine is called with V4 = 2 and V5 = 4, the result will be: U3\$ = "XXX02XXX".

The subroutine can be called repeatedly to build up more complex strings. If called with V4 greater than 99, the two digits to be inserted are the tens digit and the ones digit. If V4 is less than zero, an appropriate multiple of 100 is added in order to generate a number between 0 and 99.

### 9.3 NUMERIC UNPACK

#### Operation:

Extracts an integer from a given character string U3\$.

#### Input:

U3\$ = a character string with at least one character in it.

#### Output:

V3 = a numeric equivalent to the character string, and  
V2 = the number of the first non-numeric column in U3\$.

#### Example:

If U3\$ = "012/405", the subroutine would return the values of V3 = 12  
and V2 = 4.

If the first column of U3\$ is non-numeric (e.g., "-024"), the subroutine returns V3 = 0, V2 = 1. If the entire string U3\$ is numeric, then V2 is equal to one plus the length of the string. For the purposes of this subroutine, only the digits "0" through "9" are considered to be numeric. If a negative sign is to be considered, it must be done externally.

## 9.4 CONVERT CHARACTER DATE TO MONTH-DAY-YEAR

### Operation:

Converts character string U3\$ ("int d int d int") into three integers U1, U2 and U3. Int refers to zero or more valid numerics as defined in the previous subroutine. d is a delimiter which can be any character other than a digit. Examples of delimiters are "/", ",", " ", etc.

### Input:

Character string U3\$.

### Output:

Three integers U1, U2 and U3.

### Example:

If U3\$ = "2/14/73", the subroutine returns U1 = +2; U2 = +14; and U3 = +73.

If any integer is null (i.e., no valid numerics, which would mean two consecutive delimiters, the appropriate result is zero.

### Examples of U3\$ Conversion:

<u>U3\$</u>	<u>U1</u>	<u>U2</u>	<u>U3</u>
1/2/71	1	2	71
12-1-5	12	1	5
1248//6	1248	0	6
123/4/5	0	123	4
/1	0	1	0
1248/6	1248	6	0

## 9.5 CONVERT MONTH-DAY-YEAR TO YEAR-DAYS

### Operation:

Calculates the number of days since the beginning of the given year.

### Input:

U1 = month, U2 = day, U3 = year.

### Output:

U4 = number of days from beginning of the year U3 up to the given month and day.

### Example:

U1 = 8, U2 = 4, U3 = 71 produces U4 = 216

Leap year is considered for every day after February 28 on a year divisible by four (4). Since only a two-digit "year" is used, leap year is defined to occur when those two digits constitute a number divisible by four.

Two special circumstances should be noted. If U1 is larger than 12 or less than 1, the date must be adjusted. In this case, the appropriate multiple of 12 is added or subtracted from U1, and U3 is adjusted accordingly. Thus, the 14th month of 72 is changed to the second month of 73. Secondly, if U2 is larger than the number of days in the month U1, no special action is taken. Thus, January 40 yields U4 = 40, the same as February 9. If occurring in December, it could result in a days figure greater than 366, an undesirable possibility.

## 9.6 CONVERT YEAR-DAY TO JULIAN DATE

### Operation:

Converts the Julian date from the given year U3 and the number of the day U4 within that year.

### Input:

U3 = year, and  
U4 = days within that year.

### Output:

U0 = the Julian date. .

### Example:

U3 = 73, U4 = 321 produces U0 = 73321.

Special care should be exercised that U4 is not larger than 999, since the carry is added to the year. For the date to make sense, U4 should not be larger than 366 for a leap year and 365 for a non-leap year.

#### 9.7 CONVERT JULIAN DATE TO YEAR-DAYS

##### Operation:

Converts the given Julian date U0 to corresponding year U3 and days in that year U4.

##### Input:

U0 = Julian date.

##### Output:

U3 = year, and  
U4 = day of year.

If the variable U0 is negative or zero, the results are meaningless.

#### 9.8 CONVERT YEAR-DAYS TO MONTH-DAY-YEAR

##### Operation:

Converts a date of the form year, day of year into the form month, day, year.

##### Input:

U3 = year, and  
U4 = day of year.

##### Output:

U1 = month,  
U2 = day of month, and  
U3 = year (same as input).

##### Example:

If U3 = 73 and U4 = 216, results are U1 = 8, U2 = 4, (and U3 = 73).

If U4 is between 1 and 366, this routine works for both leap years and non-leap years. If U4 is zero or negative, U1 is set to 1 and U2 set equal to U4. If U4 is larger than the days of the current year, U2 is set equal to the excess plus 31, and U1 is set equal to 12.

## 9.9 CONVERT MONTH-DAY-YEAR TO CHARACTER FORM

### Operation:

Given a month, day and year (U1, U2, U3), creates an 8-character string (U3\$) equal to "MM/DD/YY".

### Input:

U1 = month,  
U2 = day, and  
U3 = year.

### Output:

U3 = "MM/DD/YY" where

MM = U1,  
DD = U2, and  
YY = U3.

This subroutine does not pay attention to the meaning of U1, U2 and U3. It treats them as three numbers to be combined. Accordingly, there is no check for reasonableness of the date produced. Each is assumed to be non-negative. The decimal portion, if any, is truncated and the last two digits are inserted according to the format listed above. If any is negative, an appropriate multiple of 100 is added until it becomes positive.

## 9.10 CONVERT FROM YEAR-DAY TO DAYS

### Operation:

Calculates the number of days from December 31 of the year zero to the given day of the given year.

### Input:

U3 = year, and  
U4 = day number within current year.

### Output:

U5 = total days from 12/31/00 to the U4th day of year U3.

This routine provides a way to subtract two dates.



## 9.11 ADJUST YEAR-DAYS TO NORMAL FORM

### Operation:

Given any Julian date, this subroutine adjusts the date to normal form, where normal form is YY DDD and  $D < DDD \leq 365$  (or  $D < DDD \leq 366$  when YY specifies a leap year).

### Input:

U3 = year, and

U4 = number of days displacement from the beginning of that year (positive, negative or zero...any magnitude).

### Output:

U3 = adjusted year, and

U4 = adjusted days within year.

### Example:

U3 = 71, U4 = 375 results in U3 = 72, U4 = 10, and

U3 = 70, U4 = -1086 results in U3 = 67, U4 = 10.

All the intervening leap years during the adjusting are accounted for automatically. This makes a useful tool for adding or subtracting a number of days from a date. In order to be compatible with the other date subroutines, U4 is "one" for January 1, not "zero". Thus it is the number of days since December 31 of the previous year.

## 9.12 ACCEPT DATE

### Operation:

Asks operator for today's date and converts to Julian form, displays it back to operator for verification, and stores it in form suitable for use by STANDARDS UTILITIES.

### Input:

As requested on CRT.

### Output:

Q1\$ = today's date in character form, and  
Q1 = today's date in Julian form.

This subroutine should be called before any reference to output files via the utilities '240 and '241.

## 9.13 VARIABLES

Q1	TODAY'S DATE IN JULIAN FORM
U\$10	DIGITS 0-9 (U\$='0123456789')
U(12)	DAYS FROM BEGINNING OF YEAR TO BEGINNING OF MONTH (LEAP YEAR)
U0	TODAY'S DATE IN JULIAN FORM
U0\$1	SCRATCH VARIABLE
U1	MONTH (E.G., 1=JANUARY)
U2	DAY OF MONTH
U3	YEAR (E.G., 75=1975)
U3\$8	CHARACTER STRING(E.G., '02/30/77')
U4	DAY OF YEAR (365 -OR 365 FOR LEAP YEAR)
U5	DAYS SINCE YEAR 0 (IN THIS CASE, THIS IS SINCE DEC. 31, 1900)
V	SCRATCH VARIABLE
V2	SCRATCH VARIABLE
V3	SCRATCH VARIABLE
V4	SCRATCH VARIABLE
V5	SCRATCH VARIABLE

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.

700-3459A

TITLE OF MANUAL: **System 2200 General Utilities Volume II**

COMMENTS:

\_\_\_\_\_  
Fold

\_\_\_\_\_  
Fold



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.**  
836 NORTH STREET  
TEWKSBURY, MASSACHUSETTS 01876

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Attention: Marketing Department

Fold

Cut along dotted line.



1

2



3

4

5



**WANG LABORATORIES  
(CANADA) LTD.**

49 Valleybrook Drive  
Don Mills, Ontario M3B 2S6  
TELEPHONE (416) 449-2175  
Telex: 069-66546

**WANG EUROPE, S.A.**

Buurtweg 13  
9412 Ottergem  
Belgium  
TELEPHONE 053/704514  
Telex: 26077

**WANG DO BRASIL  
COMPUTADORES LTDA.**

Rua Barao de Lucena No. 32  
Bota Fogo. ZC-01, 20,000  
Rio de Janeiro, Brazil  
TELEPHONE 226-4326  
Telex: 2123296

**WANG COMPUTERS  
(SO. AFRICA) PTY. LTD.**

Corner of Allen Rd. & Garden St.  
Bordeaux, Transvaal  
Republic of South Africa  
TELEPHONE 486-123

**WANG INTERNATIONAL  
TRADE, INC.**

836 North Street  
Tewksbury, Massachusetts 01876  
TELEPHONE (617) 851-4111  
TWX 710-343-6769  
Telex: 94-7421

**WANG SKANDINAVISKA AB**

Fredsgatan 17, Box 122  
S-172 23 Sundbyberg 1, Sweden  
TELEPHONE 08-98-1245  
Telex: 11498

**WANG NEDERLAND B.V.**

Damstraat 2  
Utrecht, Netherlands  
(030) 93-09-47  
Telex: 47579

**WANG PACIFIC LTD.**

902-3, Wong House  
26-30 Des Voeux Road, West  
Hong Kong  
TELEPHONE 5-435229  
Telex: 74879

**WANG INDUSTRIAL CO., LTD.**

110-118 Kuang-Fu N. Road  
Taipei, Taiwan  
TELEPHONE 784181-3  
Telex: 21713

**WANG GESELLSCHAFT M.B.H.**

Formanekgasse 12-14  
A-1190 Vienna, Austria  
TELEPHONE 36.32.80  
Telex: 74640

**WANG S.A./A.G.**

Markusstrasse 20  
CH-8042 Zurich 6  
Switzerland  
TELEPHONE 41.1.26.6886  
Telex: 59151

**WANG COMPUTER PTY. LTD.**

25 Bridge Street  
Pymble, NSW 2073  
Australia  
TELEPHONE 449-6388  
Telex: 25469

**WANG ELECTRONICS LTD.**

1 Olympic Way, 4th Floor  
Wembley Park,  
Middlesex, England  
TELEPHONE 01/903/6755  
Telex: 923498

**WANG FRANCE S.A.R.L.**

Tour Galleni, 1  
78/80 Ave. Galleni  
93170, Bagnolet, France  
TELEPHONE 33.1.3601212  
Telex: 68958

**WANG LABORATORIES GMBH**

Moselstrasse 4  
6000 Frankfurt AM Main  
West Germany  
TELEPHONE (0611) 252061  
Telex: 04-16246

**WANG COMPUTER SERVICES**

836 North Street  
Tewksbury, Massachusetts 01876  
TELEPHONE (617) 851-4111  
TWX 710-343-6769  
Telex: 94-7421

24 Mill Street  
Arlington, Massachusetts 02174  
TELEPHONE (617) 648-8550

**WANG**

LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876. TEL (617) 851-4111. TWX 710 343-6769. TELEX 94-7421

Printed in U.S.A.  
700-3459A  
6-75-2.5C  
Price \$10.00