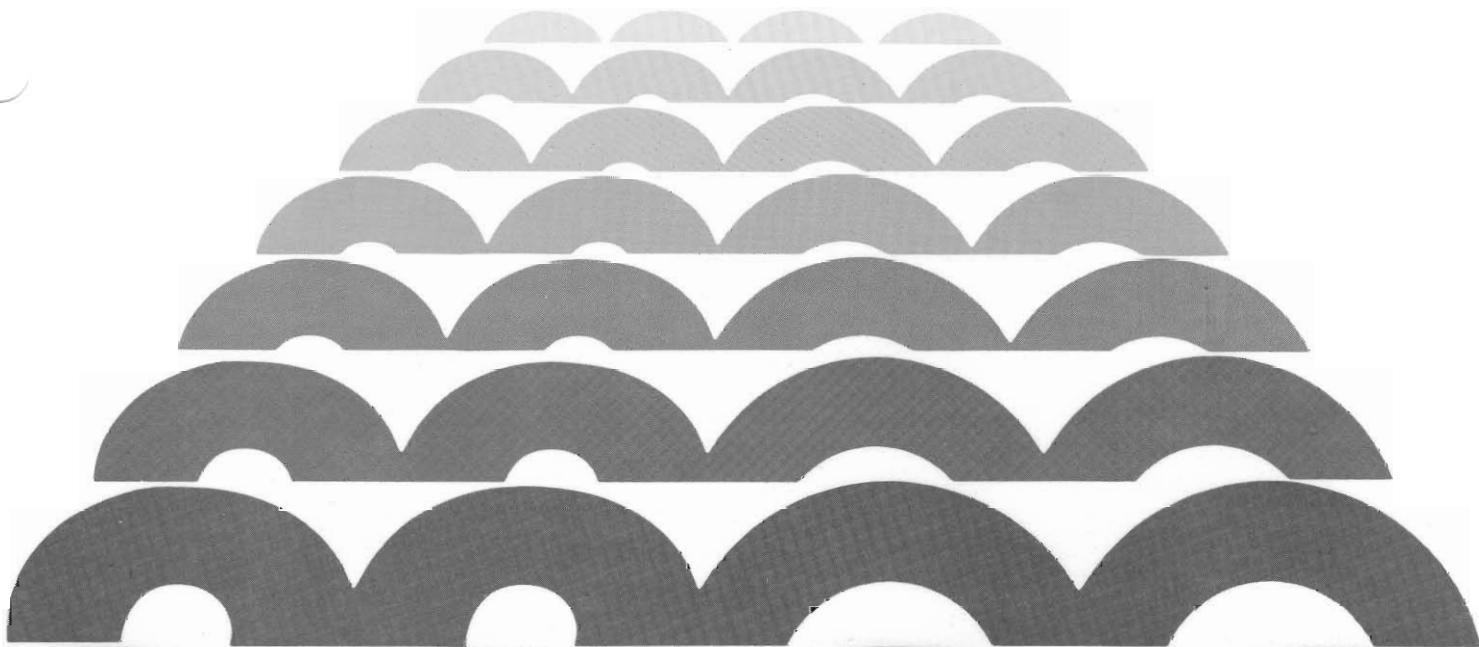


**WANG**

# **BASIC-2 Utilities Reference Manual**



# **2200**



# **BASIC-2 Utilities Reference Manual**

**1st Edition — June, 1981**  
**Copyright © Wang Laboratories, Inc., 1981**  
**700-6855**



**LABORATORIES, INC.**

---

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

## **Disclaimer of Warranties and Limitation of Liabilities**

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

### **NOTICE:**

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



LABORATORIES, INC.

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

## PREFACE

This manual describes the 2200 BASIC-2 utilities provided as a subset of the 2200 Operating System. The 2200 system utilities are programs that perform support and maintenance functions commonly used for system control and file maintenance.

Chapter 1 consists of an introduction to the BASIC-2 utility software, including an overview of possible system configurations utilities. Chapter 2 describes the procedures for master initializing the 2200 Series computer system. Information is also provided in Chapter 2 for performing system diagnostics. Specifications and operating instructions for the Partition Generator utility are provided in Chapter 3. Chapter 4 covers the process of loading the system utilities. The Partition Status utility is detailed in Chapter 5. An overview of the utility used to install new system files is provided in Chapter 6. The Format Disk Platter utility is described in Chapter 7, including information regarding both the disk and diskette. Chapter 8 details the Vertical Format Control utility. Chapter 9 discusses several aspects of system backup, including such topics as when to backup, backing up to a disk or diskette of a different size, and running the Backup and Recover utilities. Chapter 10 discusses the Move File utility and the subject of software transportation between systems.

This manual should be used in conjunction with the following manuals.

- 2200VP Introductory Manual (700-4082A)
- 2200LVP Introductory Manual (700-6164)
- 2200MVP Introductory Manual (700-4693C)
- Wang BASIC-2 Language Reference Manual (700-4080D)
- Wang BASIC-2 Disk Reference Manual (700-4081F)
- The Model 2273 Band Printer User Manual (700-5726)



## CONTENTS

CHAPTER 1	INTRODUCTION	
1.1	System Configurations .....	1-1
1.2	Overview of the Utilities .....	1-2
	Configuration Utilities .....	1-2
	Disk Maintenance Utilities .....	1-2
	Printer Utilities .....	1-3
CHAPTER 2	MASTER INITIALIZING THE SYSTEM	
2.1	Master Initialization .....	2-1
2.2	When Master Initialization Is Required .....	2-1
2.3	The Process of Master Initialization .....	2-2
	Power On .....	2-3
	Mounting the System Platter .....	2-3
	Loading the System Programs .....	2-6
2.4	Running the System Diagnostics .....	2-8
	Loading the Diagnostics Menu .....	2-9
	The CPU Diagnostic .....	2-9
	The Control Memory Diagnostic .....	2-9
	The Data Memory Diagnostic .....	2-9
CHAPTER 3	THE BASIC-2 MULTI-USER PARTITION GENERATOR	
3.1	Overview of the @GENPART Program .....	3-1
3.2	Partition Generation (System Configuration) .....	3-1
	System Configuration Parameters the User Can Specify ....	3-1
	The Process of System Configuration Using @GENPART .....	3-2
3.3	@GENPART Operating Instructions .....	3-3
	@GENPART Options .....	3-3
3.4	Generating a Sample Configuration .....	3-5
	Load a Configuration (SF'08) .....	3-6
	Clear Partitions (SF'00) .....	3-6
	Edit Partitions (SF'04) .....	3-7
	Edit Device Table (SF'05) .....	3-9
	Broadcast Message (SF'06) .....	3-10
	Save Configuration (SF'09).....	3-10
	Execute Configuration (SF'15) .....	3-11
	Delete a Configuration (SF'10) .....	3-11
	Customizing @GENPART .....	3-11
CHAPTER 4	LOADING THE SYSTEM UTILITIES	
4.1	Loading the Utility Software .....	4-1

## CONTENTS (continued)

CHAPTER 5	THE PARTITION STATUS UTILITY	
5.1	An Overview .....	5-1
5.2	The Partition Status Display .....	5-1
CHAPTER 6	INSTALL SYSTEM FILES UTILITY	
6.1	An Overview .....	6-1
6.2	Installing System Files From a New Release Diskette .....	6-1
6.3	Running the Utility .....	6-2
CHAPTER 7	FORMAT DISK PLATTER UTILITY	
7.1	An Overview .....	7-1
	Formatting the DSDD Diskette .....	7-1
	Formatting the Fixed Disk .....	7-2
7.2	Running the Utility .....	7-3
CHAPTER 8	THE VERTICAL FORMAT CONTROL UTILITY	
8.1	An Overview .....	8-1
8.2	Running the Utility .....	8-2
CHAPTER 9	DISK BACKUP	
9.1	An Overview .....	9-1
	When to Backup .....	9-1
9.2	Fixed/Removable Platters of Same Size .....	9-2
	COPY Statement .....	9-2
	MOVE Statement .....	9-3
	COPY/VERIFY Utility .....	9-3
9.3	Systems with Fixed Platters Larger than Removable Platters .	9-4
9.4	Backup Utility .....	9-4
	Files Created by the Utility .....	9-4
9.5	Running the Backup Utility .....	9-5
9.6	Recover Utility .....	9-7
9.7	Running the Recover Utility .....	9-8
	Recovering Specific Files .....	9-9
	Recovering All Active Files .....	9-9
	Recovering Entire Disk .....	9-10
	Error Messages .....	9-10



CHAPTER 10    TRANSPORTING SOFTWARE

10.1	An Overview .....	10-1
10.2	The Media .....	10-1
10.3	@MOVEFIL 3741 Format .....	10-2
10.4	@MOVEFIL Restrictions .....	10-2
10.5	Multi-volume Files .....	10-2
	Output Files .....	10-2
	Input Files .....	10-3
	Multi-volume File Format .....	10-3
10.6	Running the Utility .....	10-3

## FIGURES

Figure 2-1	Model 2270/2270A Diskette .....	2-4
Figure 2-2	DSDD Diskette .....	2-4
Figure 2-3	The System 2200 Menu .....	2-7
Figure 2-4	The Diagnostics Menu .....	2-9
Figure 4-1	BASIC-2 System Utilities Menu .....	4-1
Figure 5-1	Sample Partition Status Display .....	5-1
Figure 7-1	Disk Platter Format Utility Screen .....	7-3
Figure 8-1	Initial DAVFU Screen .....	8-2
Figure 8-2	DAVFU Control Screen .....	8-4
Figure 9-1	Backup Utility Screen .....	9-7
Figure 9-2	Recovery Utility Screen .....	9-8
Figure 10-1	Move File Utility Screen .....	10-4

## TABLES

Table 1-1	BASIC-2 Operating Systems .....	1-1
Table 2-1	Functions of the User Hardware Diagnostics .....	2-8

CHAPTER 1  
INTRODUCTION

1.1 SYSTEM CONFIGURATIONS

The utilities described in this manual are available as part of the operating systems of the 2200SVP and 2200VP or the 2200LVP and 2200MVP. The majority of the utilities are available on all systems; however, the Partition Generator and Partition Status utilities are available only for the 2200LVP and 2200MVP. The Partition Generator and Partition Status utilities are available only on the system platter that includes the 2200MVP (Multi-programming) Operating System, BASIC-2 language processor, and system diagnostics.

The VP (Single-user) BASIC-2 Operating System is used on the 2200VP and 2200SVP systems. The MVP (Multi-user) BASIC-2 Operating System is used on the 2200MVP and 2200LVP systems. However, VP (Single-user) BASIC-2 can be loaded and used with 2200MVP or 2200LVP systems.

Table 1-1. BASIC-2 Operating Systems

OPERATING SYSTEM	2200 CPU	MEDIA TYPE*
BASIC-2 (Single-user)	2200VP	SSSD Diskette
	2200SVP	DSDD Diskette
	2200MVP	SSSD Diskette
	2200LVP	DSDD Diskette
BASIC-2 (Multi-user)	2200MVP	SSSD Diskette
	2200LVP	DSDD Diskette

\* SSSD = Single-Sided, Single-Density Diskette  
DSDD = Dual-Sided, Double-Density Diskette

## 1.2 OVERVIEW OF THE UTILITIES

### Configuration Utilities

- . @GENPART: Partition Generator (for use with the BASIC-2 (Multi-user) Operating System).

@GENPART creates, saves, and executes system configurations that divide the 2200 resources among the system users. (Refer to Chapter 3.)

- . @PSTAT: Partition Status (for use with the BASIC-2 (Multi-user) Operating System).

@PSTAT displays a table indicating the current status of each partition in the current system configuration. (Refer to Chapter 5.)

- . @MENU: Program Menu

@MENU provides a menu structure for program selection. Multiple levels of menu can be set up with each successive screen displaying the next menu mode. (@MENU is not displayed as an option from the System Utilities menu; it must be loaded separately.)

### Disk Maintenance Utilities

- . @FORMAT: Format Disk Platter

@FORMAT formats software formattable disk platters, such as 2260C, 2260BC, 2280 platters; dual-sided, double-density diskettes; and 2200LVP fixed platters. (Refer to Chapter 7.)

- . @INSTALL: Install System Files

@INSTALL allows the installation of system files contained on New Releases of the 2200 Operating System Diskettes. (Refer to Chapter 6.)

- . @MOVEFIL: Move File

@MOVEFIL provides a general file move capability. Specified files or all active files are moved from one platter to another; if necessary, a file can span more than one output platter. This utility can create a new file, change the new file size of the output file, rename a file, and overwrite an existing file. File data can be recorded in 3741 format for software transport between VP/MVP and SVP/LVP systems. (Refer to Chapter 10.)

. @BACKUP: Backup Platter

@BACKUP provides the ability to copy the contents of a single disk platter to another platter or to a series of smaller platters. The primary purpose of @BACKUP is to allow the 2-, 4-, or 8-megabyte fixed disks on the 2200LVP or 2200SVP to be backed up onto several 1-megabyte dual-sided, double-density (DSDD) diskette platters. (Refer to Chapter 9.)

. @RECOVER: Recover from Backup

@RECOVER is the complementary utility to @BACKUP. The contents of the platters produced by @BACKUP are retrieved and written onto another platter by @RECOVER. This utility can only be used to retrieve information from a platter created by @BACKUP. (Refer to Chapter 9.)

### Printer Utilities

. @DAVFU: Vertical Format Control

@DAVFU allows the user to define, edit, load and test the vertical format data files for the Direct Access Vertical Format Unit (DAVFU) without the difficulty of programming long code sequences. It may be used on any Wang printer that provides DAVFU capability (e.g., 2273 Band Printer, 2235 Matrix Printer). (Refer to Chapter 8.)



## CHAPTER 2 MASTER INITIALIZING THE SYSTEM

### 2.1 MASTER INITIALIZATION

Master Initialization is the process of powering up components of the system, loading the operating system, and exercising the Central Processing Unit (CPU) to determine if any malfunctions exist.

In the 2200 Series of systems, a small area of the control memory called the "bootstrap" area, contains the only resident (hardwired) microcode. The bootstrap area is the only portion of control memory that is functional as soon as the CPU is turned on. Neither user memory nor the remainder of control memory contains any information when the system is first powered up. Before any operations can be performed by the system, the operating system and BASIC-2 interpreter must be loaded from the system platter into the remainder of control memory. The BASIC-2 interpreter translates and executes the BASIC text, oversees variable and program storage in user memory, and controls input/output (I/O).

The bootstrap performs the functions of initial CPU and memory checkout and loading the operating system. The bootstrap routines are invoked automatically whenever the system is powered on. Essentially the operator's job is to make sure that the System Platter is properly mounted in a diskette drive, or that the system files are available on a fixed disk. The user must also instruct the bootstrap routine, via the RESET button and Special Function keys, which disk address to access and which system program(s) to load.

The 2200LVP and 2200MVP can be master initialized only at Terminal 1, the terminal attached to the primary 2236MXD Terminal Processor at Port 1. For this reason, Terminal 1 should be located at the central system site as near as possible to the CPU and system disk drive.

### 2.2 WHEN MASTER INITIALIZATION IS REQUIRED

Both user memory and control memory, except for the bootstrap area, are cleared when the system is powered down. Therefore, the system must be master initialized following any power-down operation or whenever the system resources are to be reallocated. Occasionally, it may be necessary to power off and master initialize to recover from an error condition. Always save any desired programs or data residing in memory prior to powering down the system.

Because control memory and user memory are separate, it is possible to clear user memory without affecting control memory, with a CLEAR command. It is not necessary to master initialize the system following execution of a CLEAR command, because the contents of control memory are not destroyed.

### 2.3 THE PROCESS OF MASTER INITIALIZATION

The process of master initializing the 2200 system is summarized in the following four steps. Each step of the Master Initialization Procedure is described in detail in the sections that follow.

1. Power up the CPU, the system disk, each terminal, and any other device to be used.

If the system is already running and Master Initialization is required to reconfigure the system, it is not necessary to power down and then power up the system components. The Immediate mode form of \$INIT (\$INIT "password"), when executed at Terminal 1, causes the system to initiate the bootstrap routines as if it had just been powered up.

2. Mount the system platter, which contains the BASIC-2 interpreter and operating system, as well as a variety of hardware diagnostics and other system programs. The system files may be resident on a fixed disk platter, if they have previously been moved there from the release diskette using @INSTALL.
3. Use the RESET and Special Function keys to initiate loading of the system software from the system platter.
4. Select the desired software to load from the menu displayed.



## Power On

To power up the system, the user must follow these steps.

1. Turn on the CPU POWER switch.
2. Turn on the Cathode Ray Tube (CRT), printers and disk units. As soon as the CRT warms up (15 seconds), the CRT displays a self-identification message of the following form (not displayed on older versions of the terminal such as 2226 and 2236D).

```
2236DE R01 19200BPS 8 + 0 (USA)
```

where:

2236DE is the terminal model number.

R01 is the revision number of terminal firmware.

19200BPS is the data rate.

8 + 0 is the number of data bits (7 or 8); E is even parity, O is odd parity, N is no parity.

(USA) is the version of the keyboard and CRT character set.

3. Press the RESET key.
4. The following message should then appear on the CRT display of Terminal 1.

```
MOUNT SYSTEM PLATTER  
PRESS RESET
```

If the MOUNT SYSTEM PLATTER/PRESS RESET message does not appear or if it is incomplete, an error has been detected automatically by the bootstrap diagnostics.

5. Master initialize the system again. If Master Initialization cannot be performed, refer to the appropriate CPU introductory manual for error recovery procedures. If recovery is not accomplished through these procedures, call your Wang Service Representative.

## Mounting the System Platter

1. Remove the system diskette from the envelope; check to make certain that the diskette moves freely in its jacket and the write-protect notch in the diskette jacket is uncovered. For 2270/2270A diskettes, refer to Figure 2.1. For dual-sided, double-density (DSDD) diskettes, refer to Figure 2.2. When the notch is uncovered, no information can be written on the diskette, therefore, the diskette is protected against accidental destruction.

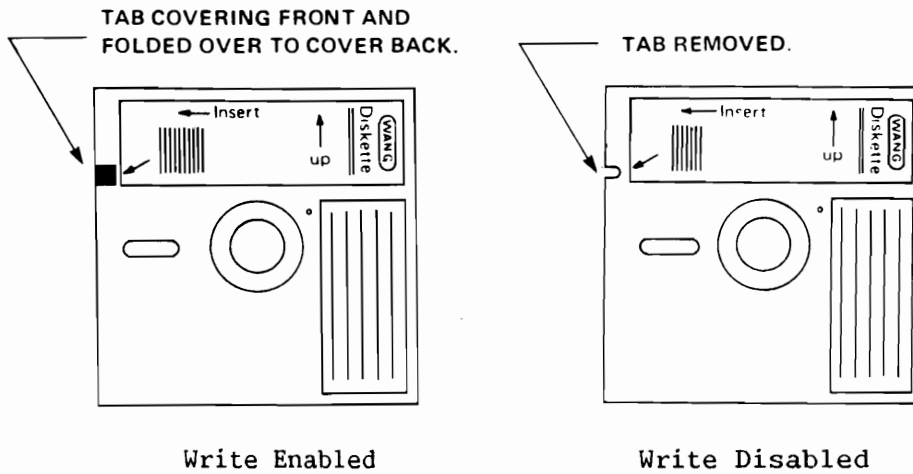


Figure 2-1. Model 2270/2270A Diskette

Figure 2-1 shows one diskette with the write-protect notch covered and another diskette with the notch uncovered. Writing on the diskette is permitted (enabled) when the write protect notch is covered and not permitted (disabled) when the write-protect notch is uncovered. The system diskette's write-protect notch should be uncovered.

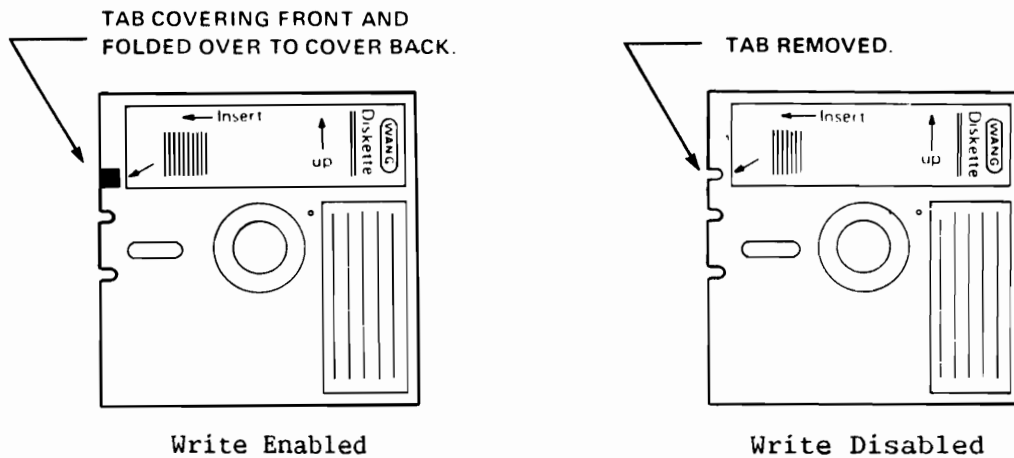


Figure 2-2. DSDD Diskette

Figure 2-2 illustrates one diskette with the write-protect notch covered and another diskette with the notch uncovered. Writing on the diskette is permitted (enabled) when the write-protect notch is covered and not permitted (disabled) when the write-protect notch is uncovered. The system diskette should be uncovered.

2. Open the door of the diskette drive by pressing the door latch immediately to the left of the door. The door will spring open.
3. Insert the system diskette into the drive, paying careful attention to the arrows on the diskette jacket that indicate the proper orientation of the diskette for insertion. (If the diskette is not mounted properly, an error will be signalled when the system attempts to read from it.) Push the diskette into the drive until it catches and holds in the slot.

CAUTION

Do not attempt to remove the diskette manually once it is inserted; serious damage to both the drive and the diskette can result.

4. Close the diskette drive door.
5. Once the System Software is loaded, the system diskette should be removed and stored in its envelope. To remove the diskette, press the release button to open the diskette drive door and eject the diskette.

NOTE

As a safety feature, the front access door of the DSDD diskette drive cannot be opened when a diskette is in the drive and the heads are engaged or when the diskette is in the process of being loaded. This door remains locked for 5 seconds after a diskette has been accessed by the system through a command (e.g., a LISTDC command). Pressing the RESET key following such a command overrides this delay and permits immediate diskette removal.

## Loading the System Programs

1. When the system platter is properly mounted, press the RESET key, which is located in the upper-left corner of the keyboard. The following prompt is displayed at Terminal 1.

KEY SF' ?

2. A Special Function key must now be pressed to specify the disk address where the system files are located. The following disk address options are available.

SF Key '00 = Load from Disk Address D11 (or 310)

SF Key '01 = Load from Disk Address D10 (or B10)

SF Key '02 = Load from Disk Address D21 (or 320)

SF Key '03 = Load from Disk Address D20 (or B20)

SF Key '04 = Load from Disk Address D31 (or 330)

SF Key '05 = Load from Disk Address D30 (or B30)

### NOTE

The bootstrap routines also allow the default file name "@@" to be overridden by entering the file name immediately before pressing the Special Function key. The name of the file to load must begin with "@" and can be no more than four characters in length. This feature allows more than one operating system to reside on a platter.

3. If the wrong Special Function key was pressed (e.g., if the system platter is mounted at Disk Address D10, but the operator presses Special Function key '00), an error message of the following form is displayed.

```
*** SYSTEM ERROR (DISK 00XX) ***  
PRESS RESET
```

4. To recover, simply press RESET and then press the correct Special Function key.
5. If pressing RESET fails, turn the system off, wait at least 5 seconds, turn the system on again, and repeat the Master Initialization sequence. If this approach fails, refer to the appropriate CPU manual for information on system errors and recovery procedures. If Master Initialization still cannot be completed, call your Wang Service Representative.

Pressing the appropriate Special Function key causes the file "@@" to be loaded. The file "@@" is a system software menu, as shown in Figure 2-3, that displays the option of loading either the BASIC-2 interpreter and operating system, or the system diagnostics.

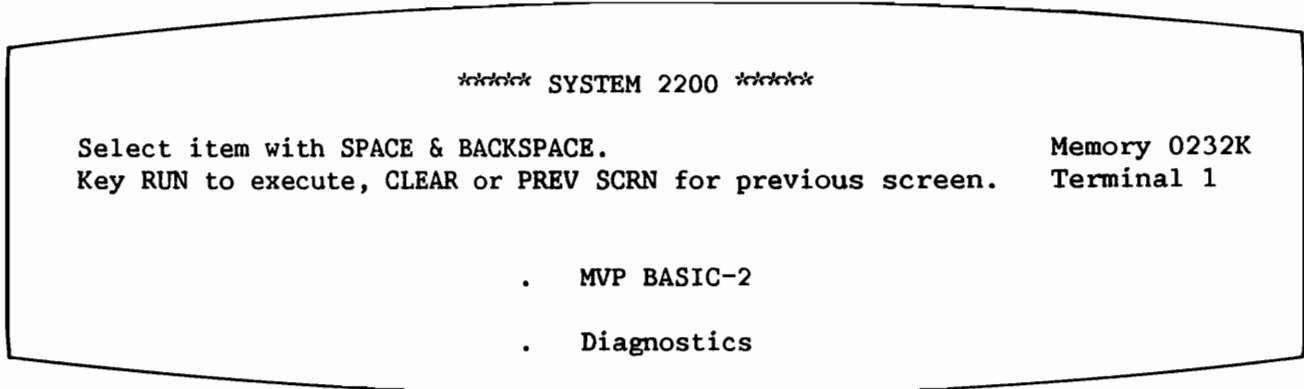


Figure 2-3. The System 2200 Menu

8. To load the multi-user BASIC-2 interpreter, select MVP BASIC-2, and press RUN to load. The following message is displayed:

Loading: MVP (Multi-user) BASIC-2, Release X.X

where X.X is the release number of the Operating System being used.

A single-user partition is set up with device addresses 215, 3X0, and BX0 automatically assigned; 3X0 and BX0 are the fixed and removable system disk platters.

If the VP Operating System is being loaded, then the menu appears with VP BASIC-2 instead of MVP BASIC-2. Therefore, to load the single-user BASIC-2, select VP BASIC-2, and press RUN to load. The READY (BASIC-2) message is displayed and the system is now available for use.

9. After the MVP BASIC-2 interpreter is loaded, the system automatically loads and runs the @GENPART Partition Generation program from the system platter. Once @GENPART has been loaded, the user can interactively create a system configuration. Alternatively, @GENPART can be customized to automatically execute a designated configuration stored on the system platter. Refer to Chapter 3.

@GENPART also allows other programs resident on the system disk to be loaded into the partitions and executed automatically when the system is configured, with no further operator intervention.

10. If @GENPART is not on the system disk, the system is configured into a limited system in which Terminal 1, the default disk drive, and any terminal printer attached to Terminal 1 are ready for limited use. The READY (BASIC-2) message is then displayed. The other terminals and system resources can be utilized (i.e., are enabled) only after configuring the system.

2.4 RUNNING THE SYSTEM DIAGNOSTICS

CAUTION

Since the diagnostic programs destroy the contents of user memory, the user should save all valuable programs and data on disk prior to running the diagnostics.

As with Master Initialization, the hardware diagnostics on a 2200MVP or 2200LVP can only be initiated at Terminal 1. The hardware diagnostics are a set of programs that exhaustively test the CPU hardware components and attempt to identify any malfunctions. Hardware diagnostics can only be run after Master Initialization or the occurrence of a system error. Each diagnostic program runs continuously until RESET is pressed or an error is detected. When an error is detected, diagnostic processing stops or an error message is displayed. For a discussion of possible error causes and recovery procedures, refer to the appropriate CPU manual.

A hard copy listing of the diagnostic messages can be obtained by turning on the printer attached to the CPU or terminal during diagnostic execution. The printer attached to the CPU must be assigned Address 215 and the SELECT button must be illuminated. A hard copy trace of the diagnostic run should be obtained to aid in the location and correction of problems.

The hardware diagnostics should be executed:

1. At least once every 60 to 100 hours of operation.
2. Whenever errors cause a hardware malfunction to be suspected.
3. Whenever a Wang Service Representative requests their execution.

Execute the hardware diagnostic routines as quickly as possible after observing a suspected problem, to ensure that the environmental conditions of the test most closely resemble those under which the problem occurred. Refer to Table 2-1 for a functional description of each diagnostic.

Table 2-1. Functions of the User Hardware Diagnostics

Diagnostic	Function
CPU Control Memory Data Memory	Tests the 2200 Processor Tests Control Memory Tests Data Memory

## Loading the Diagnostics Menu

From the System 2200 menu displayed when master initializing (refer to Figure 2-3), select Diagnostics. The Diagnostics menu, as shown in Figure 2-4, is displayed. Each individual diagnostic can now be loaded directly from this menu.

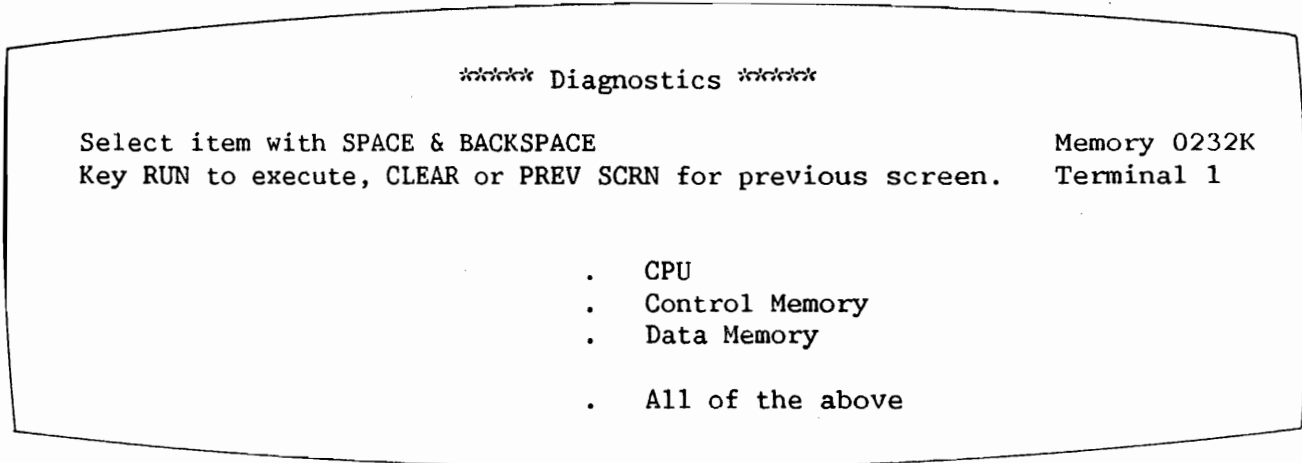


Figure 2-4. The Diagnostics Menu

### The CPU Diagnostic

The CPU diagnostic is designed to test the 2200 processor. The test runs continuously until either an error occurs (the pass number stops incrementing) or RESET is pressed. If an error occurs, call your Wang Service Representative. When a sufficient number of successful test passes have occurred (5 to 10 minutes), press RESET. The System 2200 menu can be restored by pressing RESET and the proper Special Function key to load the operating system.

### The Control Memory Diagnostic

The Control Memory diagnostic is designed to test the control memory. These control memory diagnostic tests are repeated until RESET is pressed. Error messages are displayed or printed whenever memory failures are discovered. When a sufficient number of successful test passes have occurred (at least 5 to 10 minutes), press RESET. The System 2200 menu can be restored by pressing RESET and the proper Special Function key to load the Operating system.

### The Data Memory Diagnostic

The Data Memory diagnostic is designed to test data (user) memory. The user memory tests are also repeated until RESET is pressed. Error messages are displayed or printed when memory failures are discovered. When a sufficient number of successful test passes have occurred (at least 5 to 10 minutes), press RESET. The System 2200 menu can be restored by pressing RESET and the proper Special Function key to load the operating system.





## CHAPTER 3 THE BASIC-2 MULTI-USER PARTITION GENERATOR

### 3.1 OVERVIEW OF THE @GENPART PROGRAM

The Partition Generator (@GENPART) utility available on the Multi-user BASIC-2 Operating System divides the system resources, memory and peripherals, of the 2200LVP and 2200MVP among the users on the system.

When the MVP BASIC-2 option is selected from the System 2200 menu during master initialization of the 2200LVP or 2200MVP, the system loads and runs the @GENPART program from the system platter. @GENPART allows the user to interactively create a system configuration. Alternatively, @GENPART can be customized to automatically execute a designated configuration stored on the system disk. @GENPART also allows other programs resident on the system disk to be loaded into designated partitions and executed automatically when the system is configured. Configuration definitions created by the operator may be saved on the system disk in a data file called '@SYSDFILE'.

### 3.2 PARTITION GENERATION (SYSTEM CONFIGURATION)

Partition generation (system configuration) divides the resources of the 2200 system among the various users. This section discusses the use of the @GENPART utility program to create, save, and execute system configurations.

#### System Configuration Parameters the User Can Specify

The user can specify the following ten parameters when configuring a system.

1. The number of terminals
2. The number of partitions
3. The size of each partition
4. The terminal associated with each partition
5. The programmability of each partition
6. The bootstrap program for each partition
7. The addresses of the peripherals attached to the system
8. The access to peripherals
9. The system message
10. The system reconfiguration password

## The Process of System Configuration Using @GENPART

The process of Master Initialization, as described in section 3.1, creates a limited 2200 system having a single partition with all of user memory controlled by Terminal 1. Only Terminal 1, any terminal printer attached to it, and the system disk drive are operative at this time. No other system devices are available until a configuration is executed. As a part of Master Initialization, the system microcode will automatically load and run the BASIC program file @GENPART from the system platter if such a file exists. (@GENPART is always assumed by the system to be the name of the system configuration program, whether Wang-supplied or user-written.) If @GENPART is not on the system platter, the READY (BASIC-2) message is displayed at Terminal 1.

When @GENPART is first executed, the parameters from the previous configuration (called 'current') are loaded; a list of previously saved configurations is displayed along with a prompt inquiring if a different configuration is to be loaded; and the list of @GENPART options is displayed.

The user then proceeds to enter responses to the prompts displayed by each option and uses the Special Function keys to advance from option to option. Refer to the example of this general procedure in the "Generating a Sample Configuration" section which follows.

The standard Wang @GENPART program allows users two basic options. The general procedure for each of these options follows.

### 1. Creating configurations to be executed or stored for future use.

If the user is creating a configuration for the first time, or wishes to modify a previously defined one, the Special Function keys can be used to load and modify the old configuration or to create a new definition, execute it, and/or store the configuration for future use.

### 2. Loading and executing previously defined configurations.

If the user wishes to execute a system configuration that has been previously defined and stored in the configuration file on the system platter without modifying it, he can select a configuration from the list of previous configuration names displayed on the screen and manually execute one. It is also possible to modify the @GENPART utility program so that a specified configuration is loaded and executed automatically when the system is master initialized. Section 3.5, "Customizing @GENPART", describes how to make this modification.

### 3.3 @GENPART OPERATING INSTRUCTIONS

#### @GENPART Options

SF'00 - Clear Partitions -- Clears the partition configuration parameters currently in memory; allows the user to specify the total number of terminals and the number of partitions in each bank; then advances to SF'04 (Edit Partitions). Any number of terminals from one to twelve may be specified. The number of partitions permitted in each bank can range from one to sixteen; however, the total number of partitions can not exceed sixteen. There must be at least one partition in each bank that is to be used. Memory partitions must be contiguous, i.e., there should be no memory remaining for partitions in Bank 1 before specifying memory for partitions in Bank 2. The Master Device Table is not altered when this function is selected.

SF'01 - Clear Device Table -- Clears the Master Device Table currently in memory; resets the peripheral default values to /215, /310, /320 and allocates these devices to all users; then advances to SF'05 (Edit Device Table). The default device addresses can be edited if necessary.

SF'02 - Divide Memory Evenly -- Divides the remaining memory in a bank equally among all partitions in that bank not yet allocated memory. By default, this division is performed for all banks.

SF'04 - Edit Partitions -- Displays and allows editing of partition parameters such as memory size, terminal assignment, programmability, and name of a program to be automatically bootstrapped. SF'04 does not allow addition of new partitions or deletion of defined ones for an existing configuration. The partition editing options are as follows.

Partition Size: Any memory quantity greater than or equal to 1,280 bytes (1.25K) specified in 256-byte (1/4K) increments.

Terminal Assignment: Any terminal number from 0 to 12 is valid. Terminal zero is the null terminal; a partition assigned to the null terminal is always available to any requesting terminal. Any partition may be assigned to any terminal (a terminal can support several partitions), but all partitions must be assigned to a terminal, even if they are to contain background jobs that never print on the CRT or require keyboard entry. In general, the lowest numbered partition(s) assigned to a terminal should contain the foreground (interactive) job(s) for that terminal. Background jobs should be placed in the higher-numbered partitions. Only the terminal to which a partition is assigned can list or modify the program in that partition. Although other partitions can access global program text and modify global variables, it is not possible for other partitions to list or modify the program text in a global partition.

Programmability: Any partition can be specified for Disabled Programming mode. The terminals connected to disabled programming partition(s) are inhibited from entering or modifying program text or performing a number of other system operations. Thus, the operator is prevented from inadvertent or unauthorized use of certain programs and data.

Bootstrap Programs: One or more programs that reside on the system disk, can be loaded into the partitions, and run automatically without operator intervention when a configuration is executed. This feature is particularly useful for setting up background and global partitions and forcing terminals to execute particular BASIC software. When no bootstrap program is specified for a partition, the READY (BASIC-2) display will appear on the CRT of the terminal currently attached to that partition when the configuration is executed.

SF'05 - Edit Device Table -- Displays and allows editing of the device-addresses and allocation of all peripherals attached to the system. The Master Device Table default values are read from disk and displayed on the screen. All peripherals attached to the system (other than the terminals and terminal printers attached to them) must be specified in the Master Device Table. By default, all peripheral devices are available to all partitions. However, devices can be assigned exclusively to one partition until the next system configuration is executed, by entering the number of the partition that is to have control of the device in the Master Device Table. Console device addresses, i.e., /005 CRT, /001 keyboard, /204 terminal printers, are not specified in the Master Device Table. For disk controllers that respond to more than one address, only the primary address must be specified (i.e., /310 but not /B10, et cetera). For all other multiaddress controllers, all valid addresses must be listed. For addresses that will differ by the first digit only (device type), only the normal address must be specified. The default table values are the values of the saved configuration that was last used by the system. However, if the system platter was write-protected at that time, the default values are the ones used during the previous system configuration.

SF'06 - Edit \$MSG -- Displays and allows editing of a user-defined broadcast message that will be displayed on each terminal's CRT whenever the READY message is displayed. The user-defined message is displayed on line 0 of the CRT immediately above the READY message.

SF'08 - Load Configuration -- Loads a configuration from the system configuration file on the system platter. To modify and/or execute any previously defined configuration other than the most recently executed configuration (configuration "current"), this option must be used to load the named configuration from the system platter.

SF'09 - Save Configuration -- Saves a system configuration in the system configuration file on the system platter under a user-specified name. The name has a maximum length of eight characters. If the user specifies a configuration name already used, @GENPART will verify that the user desires to replace the old configuration with the configuration currently in memory. Note that the system disk must have a write-tab (unprotected) in order to save configurations on it. The values of the Master Device Table currently stored in memory are saved to disk and may be used as default values during future Master Initializations (Refer to '05 - Edit Device Table).

SF'10 - Delete Configuration -- Deletes a configuration from the disk configuration file.

SF'15 - Execute Configuration -- Allows the operator to review and then execute a configuration. This configuration will automatically be saved in the configuration file on disk, under the name "current", when the configuration is executed if the disk platter has a tab (not write-protected). Once a configuration has been executed, the system may be reconfigured again whenever the system is powered up, or when proper execution of the Immediate mode form of the \$INIT statement occurs. The configuration scheme, except for requested partition/terminal assignment change, will remain in effect until the system is reinitialized.

#### 3.4 GENERATING A SAMPLE CONFIGURATION

The following example illustrates how @GENPART could be used to configure a system. A 2200LVP with 128K bytes of memory, three terminals, and a telecommunications option is to be configured. The configuration, named SAMPLE, will have four partitions. A 16K-byte telecommunications program will be designated for automatic bootstrapping as a background job sharing Terminal 1. Disabled programming will be specified for this partition so that it cannot be inadvertently modified. The remaining memory will be divided equally among the other three partitions.

In general, the order of executing options is as follows.

1. SF'08 to load a configuration
2. SF'00 to modify this configuration by adding or deleting partitions
3. SF'04 to create the new partition parameters
4. SF'05 to create the Master Device Table
5. SF'06 to create the broadcast message
6. SF'09 to save the configuration with a name other than 'current'
7. SF'15 to execute the configuration
8. SF'10 to delete the configuration

In the example that follows, these options are discussed in the probable order of use.

Load a Configuration (SF'08)

When @GENPART is first executed, the following display occurs without pressing SF'08.

\*\*\* WANG 2200MVP PARTITION GENERATION PROGRAM \*\*\*

LIST OF STORED CONFIGURATIONS (#PARTITIONS)  
current ( 1 )

LIST OF OPTIONS:  
SF'00 - clear partitions  
SF'01 - clear device table  
  
SF'02 - divide mem. evenly  
  
SF'04 - edit partitions  
SF'05 - edit device table  
SF'06 - edit \$MSG  
  
SF'08 - load configuration  
SF'09 - save configuration  
SF'10 - delete config.  
  
SF'15 - execute

Configuration 'current' loaded. Name of configuration to load? \_\_\_\_\_

The last configuration executed, called 'current', is automatically loaded. To load any other configuration, enter its name, then press RETURN. Since a completely new configuration is to be created, i.e., the total number of partitions in a previously defined configuration is to be modified, press SF'00 (Clear Partitions).

Clear Partitions (SF'00)

The program responds with a display that first requests the number of terminals attached to the system. Prompts requesting the operator to specify the number of partitions in each bank of memory appear next. The amount of available memory in each bank is also calculated and displayed. The figures are then automatically updated after allocation of memory for each partition. Note that in the following display, the system has subtracted the 3K bytes of system overhead from the available memory in Bank 1 and the unavailable 8K bytes from the figure displayed for Bank 2. The display appears as follows.

Available memory: 61.00 K 56.00K  
Remaining memory: 61.00 K 56.00K

No. of terminals? 3  
No. of partitions in bank 1? 2  
No. of partitions in bank 2? \_

In this display, the operator has already specified three terminals attached to the system and two partitions in the first bank. To complete this phase of partition generation, the operator should enter "2" in response to the last prompt since there will be four partitions total, and press RETURN. The program automatically invokes option SF'04 (EDIT Partitions) to allow the editing of partition parameters.

Edit Partitions (SF'04)

This option displays the default parameters for all partitions and initiates a cycle of prompts for altering these parameters. The cycle recurs continuously until another option is selected. The user can modify the parameters for each partition. The display, which is updated each time an item is entered, appears as follows.

PARTITION	SIZE (K)	TERMINAL	PROGRAMMABLE	PROGRAM
1	-	1	Y	
2	-	2	Y	
-----				
3	-	3	Y	
4	-	1	Y	

The following series of prompts is displayed in succession at the bottom of the screen.

Edit which partition (default = 1)? \_

In this example, the telecommunications program will be run in Partition 2. Begin by editing the parameters for Partition 2. Enter 2, then press RETURN. Note that an asterisk (\*) appears beside the number of the partition whose parameters are being edited. The following prompt, requesting the amount of memory to be allocated to this partition, is then displayed.

Partition Size (default = 0 K)? \_

Any value greater than 1.25K and less than the amount of remaining user memory in the bank is a valid response. If a partition in Bank 1 is specified such that it resides entirely within the universal global area, a "u" will appear next to the displayed partition number to indicate this condition. Running the telecommunications program in this partition requires 16K bytes of memory. To allocate 16K bytes of memory to Partition 2, enter 16 and then press RETURN. The following prompt is then displayed.

Terminal (default = 2)? \_

The telecommunications program will be a background job controlled by Terminal 1. To assign Partition 2 to Terminal 1, enter 1 and press RETURN. The following prompt then occurs.

Enable programming (Y or N)? \_

By default, programming is allowed for all partitions. Disabled programming will be specified for Partition 2, however, to prevent inadvertent modification of the telecommunications program. To specify Disabled Programming mode for this partition, enter N, then press RETURN. The name of a program to be automatically loaded into this partition is now requested as follows.

Name of program to load?  
-----

The name of the telecommunications program that will be run in Partition 2 is TELE-COM. Enter TELE-COM and press RETURN. When the configuration is executed, the telecommunications program TELE-COM will be automatically loaded from the system disk into Partition 2 and run.

At this point, editing of the parameters for Partition 2 is complete. Partitions 1, 3, and 4 require further modification, however. The remaining memory will be divided evenly among the remaining partitions. Press SF'02 (Divide Memory Evenly) and the system responds with the following prompt.

Divide memory evenly in which bank (default = all)?\_

Since memory is to be divided evenly in all banks, press RETURN.

The system returns to the initial EDIT WHICH PARTITION? prompt. Finally, the user must assign Terminal 2 to Partition 4. Enter this value into the table for Partition 4. Upon completion of this operation, the table display appears as follows.

PARTITION	SIZE (K)	TERMINAL	PROGRAMMABLE	PROGRAM
1	45.00	1	Y	
2	16.00	1	N	TELE-COM
-----				
3	28.00	3	Y	
4	28.00	2	Y	

Once all partitions have been edited, SF'05 (Edit Device Table) is used to leave the Edit Partitions cycle and invoke the Edit Master Device Table option. Note that it is legal to exit the Edit Partitions Cycle (SF'04) without answering all prompts.



EDIT Device Table (SF'05)

This option displays the default values in the Master Device Table, which appear on the CRT screen as follows. Note that by default, devices are available to all users.

DEVICE	PARTITION	DEVICE	PARTITION
1. /215	all	17.	
2. /310	all	18.	
3. /320	all	19.	
.			
.			
.			
.			
.			
16.		32.	

Edit which entry (default = 1)? \_

In this sample configuration a fourth device, the telecommunications controller, is used in addition to the three default devices. The device-address of this controller is /01C. To add this device to the Master Device Table, enter the number 4 and then press RETURN. An asterisk (\*) will appear beside the number 4 in the table. Several prompts are now displayed in succession on the bottom of the screen, and the table is updated each time an item is edited. First, the user is requested to enter the device-address with the following prompt.

Device address (default = /000, /000 to delete entry)? \_

Enter /01C and then press RETURN. Another prompt now appears requesting the user to allocate the device to one or all partitions.

Allocate device to which partition (default = all)? \_

Enter 2, then press RETURN to allocate the controller to Partition 2. This cycle will recur to allow the user to edit all entries in the Master Device Table. Since the parameters for all peripherals and partition allocation have been specified, the user can select another Special Function option to exit the Edit Device Table Cycle.

### Broadcast Message (SF'06)

When SF'06 (Broadcast Message) is pressed, the following display occurs at the bottom of the CRT display.

Broadcast message:  
-----

Any message where the number of characters and spaces does not exceed the number of dashes displayed on the CRT is valid. For this example, enter \* \* \* THE SYSTEM WILL GO DOWN AT NOON \* \* \* and then press RETURN. When the broadcast message has been entered, all partition-generation parameters for the sample configuration have been specified. This configuration can now be saved for later use (SF'09) or executed (SF'15). Pressing SF'09 will allow the operator to save this configuration on disk under a unique name. However, the broadcast message is not saved on disk.

#### NOTE

The system is in EDIT mode during entry of the broadcast message. While in EDIT mode, all SF keys revert to their system-defined EDIT functions. The SF keys cannot be used for their @GENPART-defined functions until the entry of the broadcast message is complete and the system leaves EDIT mode.

### Save Configuration (SF'09)

When SF'09 is pressed, the system requests a name for the new configuration by displaying the following prompt.

Check configuration to save. Configuration name? -----

#### NOTE

In order to save a configuration on the system disk, the disk must be write-enabled (i.e., the write-protect notch must be covered). An error will result otherwise.

The configuration currently in memory will automatically be saved under the name 'current' if the system disk is not write-protected. However, each time a new configuration is executed, its parameters replace the old contents in the 'current' file. A configuration should be saved under a unique name so that it can be retrieved for future use. The name to be used for the sample configuration just created is SAMPLE. Enter SAMPLE, then press RETURN. The configuration is saved under the name SAMPLE, and the edited values of the Master Device Table are saved on disk for future use as defaults.

### Execute Configuration (SF'15)

Once all parameters of a configuration have been defined, the system configuration can be executed. To execute a configuration, press SF'15. The configuration table will appear along with the following prompt.

Check configuration. OK to execute (Y or N)? \_

This prompt requests the operator to verify the configuration parameters being executed. If N is entered, the system returns to the beginning of the Edit Partitions Cycle (SF'04). If Y (RETURN) is entered, the following prompt is displayed.

Reconfiguration password? SYSTEM

The password allows the operator at Terminal 1 to reconfigure the system without powering down, while preventing unauthorized reconfiguration. Any alphanumeric string up to eight characters in length is permitted. We will change the reconfiguration password from the default value SYSTEM to some user-defined value. For example, enter SECRET. The configuration is executed and the reconfiguration password is now SECRET. In order to reconfigure the system without powering down, \$INIT "SECRET" would be entered and executed to allow the system to invoke the control bootstrap routines that are usually invoked just after the system is powered up.

### Delete a Configuration (SF'10)

Since this is only a sample configuration, delete it from the configuration file to save more space for actual configurations. To delete a configuration, press SF'10. The following prompt then requests which configuration to delete.

Delete which configuration?  
-----

Enter SAMPLE, then press RETURN, and the configuration will be deleted from the configuration file on the system disk.

### 3.5 CUSTOMIZING @GENPART

Once they have been initially defined and stored on disk, configuration parameters in a specified system configuration can be passed to the operating system and executed automatically during master initialization. The REM statements near the beginning of the @GENPART program tell the user how to modify the program to operate in this manner.

When the Wang utility @GENPART does not meet a user's needs, it is also possible to create a customized configuration program by using the BASIC-2 statement \$INIT. The discussion of \$INIT in the Wang BASIC-2 Language Reference Manual presents various methods for producing a customized configuration program.



CHAPTER 4  
LOADING THE SYSTEM UTILITIES

4.1 LOADING THE UTILITY SOFTWARE

Following system configuration, the System Utilities menu can be easily loaded. The following two commands will display this menu.

1. SELECT DISK xxx (Press the RETURN key)

The three-digit hexadecimal number xxx is the device address of the disk on which the system utilities reside.

2. LOAD RUN (Press the RETURN key)

The System Utilities Menu in Figure 4-1 is displayed. Utilities that require operator-entered information will display a series of prompts requesting this information. When prompted, type in the necessary response and press RETURN; the utility will request additional information or perform the required procedure. Each utility is handled individually in the following chapters.

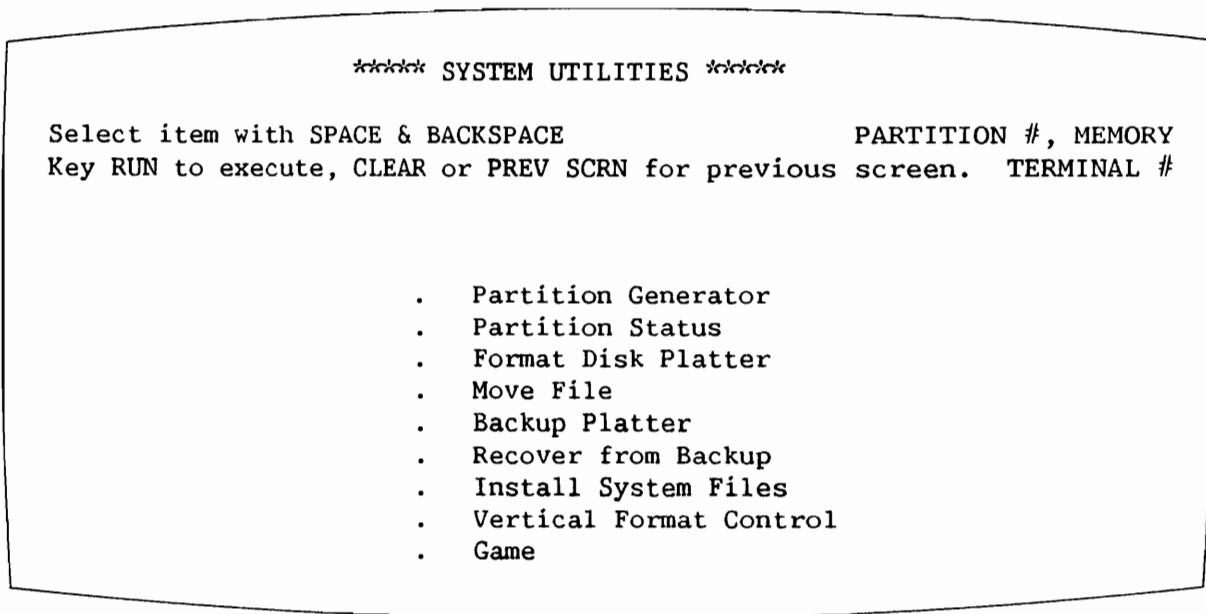


Figure 4-1. BASIC-2 System Utilities Menu



CHAPTER 5  
THE PARTITION STATUS UTILITY

5.1 AN OVERVIEW

The Partition Status utility is available from the Multi-user BASIC-2 System Utilities menu. The Partition Status program provides valuable information about each partition currently defined on the 2200LVP or 2200MVP system. This utility uses the BASIC statement \$PSTAT to obtain information about system status. A detailed description of this statement may be found in the Wang BASIC-2 Language Reference Manual.

5.2 THE PARTITION STATUS DISPLAY

From the System Utilities menu (refer to Figure 4-1), select and run the Partition Status utility. A display, similar to the example shown in Figure 5-1, appears on the screen.

***** PARTITION STATUS *****				MVP Release 2.2			Partition #				
#PART	USER-MSG	BANK	SPACEK	PRGM	#TERM	@NAME	ERR	TEXT	@P	DATA	I/O
01		01	05.00	P	00-D	MSG	00	01	01	01	00
02		01	20.00	P	01-W		00	02	02	02	00
03	POS-CASH	<u>01</u>	36.00	P	01-A		00	03	03	03	01
04		02	28.00	P	02-A		00	04	04	04	01
05		<u>02</u>	28.00	P	03-W		00	05	05	05	00
06		03	28.00	P	04-A		00	06	06	06	00
07		03	28.00		05-A		00	07	07	07	01

-----

Press SF'0 to attach terminal to a different partition.  
Press 'FN' or 'TAB' to return to menu.

Figure 5-1. Sample Partition Status Display

The Partition Status display, which conveys valuable information about each partition to the user, consists of the following information.

- #PART - the partition number
- USER-MSG - the optional Partition message
- BANK - the bank in which the partition resides
- SPACEK - the size of the partition
- PRGM - whether or not the partition is programmable.
- #TERM - the terminal currently attached to the partition, and its current state of operation.
- @NAME - whether the partition is global and if so the name
- ERR - the numeric portion of the last error encountered in the partition
- TEXT - the number of the partition containing the program text currently being executed
- @P - the number of the partition currently selected for global operations
- DATA - the number of the partition containing the data statements that the read pointer is currently pointing to
- I/O - the address of the device that the partition is communicating with or waiting for (I/O).

From the display in Figure 5-1, the following type of information regarding the current activity of each partition on the system can be obtained. For example, this display shows that there are three partitions in Bank 1. Partition 1 is a programmable partition of 5K. It is also a global partition with the global name of MSG. It has no terminal number assigned and is in a detached state, indicated by the 00-D in the #TERM column. The zero terminal number means that the partition has not requested a terminal.

The other two terminal indicators A and W, also shown in the #TERM column, refer to attached and waiting, respectively. Partition 03 is attached to terminal 1 as indicated in the #TERM column by 01-A. Partition 02 has requested a terminal and is waiting for it to be attached, as indicated by the 01-W in the #TERM column. An indicator of W exists in the case of a background job running in Partition 02 that had reached a point where a terminal is required to display a prompt or a message. Activity in that partition remains in a holding pattern until such time that a terminal is attached and the message or prompt can be displayed.



Partition 07 is defined as non-programmable, indicated by the blank in the PRGM column. Program text in this partition may not be altered.

The PSTAT display indicates to the user what jobs are waiting for a terminal. Partitions assigned to that user's terminal can then be controlled in order to provide a terminal to those jobs in a wait state. The operator attaches the terminal to the waiting partition in order to allow the background job to continue and can also release his current partition for use by another terminal. (Refer to Chapters 4 and 5 and to the \$RELEASE TERMINAL and \$RELEASE PART statements in the BASIC-2 Language Reference Manual.)

SF'0 provides the operator with the ability to attach the terminal to any available partition. Pressing SF'0 will then ask to which partition the current terminal is to be attached.

Which partition is terminal 1 to be attached to (#PART)? \_\_

Enter the partition number and press RETURN.

The operator may then release the current partition for use by another terminal. The following prompt is displayed.

Do you want to release partition X for another terminal (Y or N)? \_\_

Following the response to this prompt, the terminal will be attached to the new partition and that job continues execution. The current partition is released for another terminal (i.e., assigned to terminal 0) and the PSTAT information will be updated accordingly. At the time that the partition is released, the @PSTAT utility loads the @MENU utility into that partition. When a subsequent terminal is attached to this partition, the utilities menu is displayed on its screen.



CHAPTER 6  
INSTALL SYSTEM FILES UTILITY

6.1 AN OVERVIEW

The Install System Files utility allows for the installation of system files by moving specified system files from one disk platter to another. (NOTE: All 2200 system files have names starting with the @ character.) This utility should be invoked when updates to the operating system are received. Its purpose is to move the system files contained on a new release diskette onto one of the user's other disks.

6.2 INSTALLING SYSTEM FILES FROM A NEW RELEASE DISKETTE

Many 2200 configurations have disk drives that have larger storage capacities, or offer faster access than the standard floppy drives. For this reason, the users may want to copy the system files from the release diskette onto another disk unit. The user can thereafter load the system programs from the other disk drive. At least one copy should be made of the entire system disk in case the original is accidentally damaged or destroyed. The original diskette, either a 2270A or DSDD, can act as a backup for the system disk. After the system programs have been copied onto the fixed disk they can be recopied onto a separate diskette to create a duplicate back-up system disk.

The @INSTALL utility can be used to copy the system files from the 2270A or DSDD diskette to another disk drive. Within the @INSTALL utility is a set of DATA statements that contain the names of all the system files to be moved. With three exceptions, the utility is designed to automatically move all the system files from one disk to another. The exceptions are the 'START', '@SYSFILE', and '.STARTD' files.

The utility processes the 'START', '@SYSFILE', and '.STARTD' programs in the following manner. If these programs exist on both source and destination platters, the source file is not moved. These programs will only be moved if they do not exist on the destination platter.

The @INSTALL utility is recommended for system updates, since it overwrites only the system files being changed and leaves the remainder of the destination disk intact.

After the system files have been copied onto the fixed disk, extra backup copies of the system software can be made by running the utility again and copying from the destination platter to another platter.

### 6.3 RUNNING THE UTILITY

Select the Install System Files utility from the System Utilities menu (refer to Figure 4-1). The following is displayed.

\*\*\* SYSTEM SOFTWARE INSTALLATION UTILITY \*\*\*

Press 'FN' or 'TAB' to return to menu.

Input address = D11

The system will default to an input address of D11. Enter the proper address and press RETURN, or simply press RETURN to accept the default value of D11.

After pressing RETURN, the following message is displayed.

Output address = D10

The system defaults to an output address of D10. Enter the address of the destination platter and press RETURN, or simply press RETURN to accept D10.

The program will then display a message indicating which operating system is being installed on the destination platter. All existing system files are automatically updated. During the course of the installation, several messages are automatically displayed, showing the names of the files moved.

After the utility has successfully installed all system files, the message "Completed software installation." is displayed. To return to the System Utilities menu, press the FN or TAB key.

CHAPTER 7  
FORMAT DISK PLATTER UTILITY

7.1 AN OVERVIEW

Before data can be recorded on a disk platter, the platter must be formatted. Formatting involves assigning a unique address to each sector on the disk platter, along with certain control information which helps the system maintain the disk and check the validity of information written to and from it. Zeros are then written into the remaining area of each sector. On certain disks, formatting also certifies the platter and assigns alternate sectors to those sectors that do not meet certification specifications.

This utility will format software formattable disk platters, such as the 2260C, 2260BC, 2280 platters; dual-sided, double-density diskettes; and 2200LVP or 2200SVP fixed platters. Running this utility produces the same results as executing a \$FORMAT command, but carefully prompts the user to be sure that the proper platter is being formatted, since all previously recorded data is eliminated by the formatting process. On certain disk units (e.g., 2270, 2270A, and 2260B) formatting is initiated by pressing the format button located on the disk housing.

Formatting the DSDD Diskette

After initializing the system, the system disk drive can be used for program and data files. A blank, unused DSDD diskette, however, must be formatted before it can be used. Ordinarily, a used diskette is not formatted before storing additional files since formatting destroys any information previously recorded on the diskette. For this reason, diskettes containing packaged programs must never be formatted or scratched.

As indicated on the permanent label attached to each diskette, a diskette is protected from accidental over-writing when the write-protect notch is uncovered. Before programs or data can be written on the diskette, the notch must be covered to disable the write-protect feature.

To format the DSDD diskette, proceed as follows.

1. Remove the diskette from its envelope. Cover the write-protect notch.
2. Properly orient the diskette in the diskette drive according to the arrows on the label. Close the drive latch.

3. Press the RESET key on the keyboard.
4. To initiate disk formatting, select the Format Disk Platter utility from the System Utilities menu (refer to Figure 4-1).
5. Answer the prompts and follow the procedure described in Section 7.2, "Running the Utility".

It takes approximately one minute, 45 seconds to format a DSDD diskette.

6. If formatting is unsuccessful, a format error (ERR93) is returned. Generally errors result for one of three reasons.
  - a. The drive latch is not tightly closed
  - b. The write-protect notch is not covered
  - c. The diskette is defective
7. Remove the diskette from the drive and proceed to format another.

NOTE

If diskette cannot be formatted, it cannot be used and should be discarded.

#### Formatting the Fixed Disk

Before data can be recorded on a fixed disk platter, the platter must be formatted. To format the fixed disk, proceed as follows.

1. Be sure the disk unit is powered on and in RUN mode.
2. Press the RESET key on the keyboard.
3. To initiate disk formatting, select the Format Disk Platter utility from the System Utilities menu (refer to Figure 4.1).
4. Answer the prompts and follow the procedure described in Section 7.2, "Running the Utility".
5. If formatting is unsuccessful, a format error (ERR 93) will result. A format error may occur if the disk unit has not been properly powered on or is not in RUN mode. Turn the disk off, back on, and repeat the formatting procedure. If the error persists, call your Wang Service Representative.

## 7.2 RUNNING THE UTILITY

The screen in Figure 7-1 is displayed after selecting Format Disk Platter from the System Utilities menu.

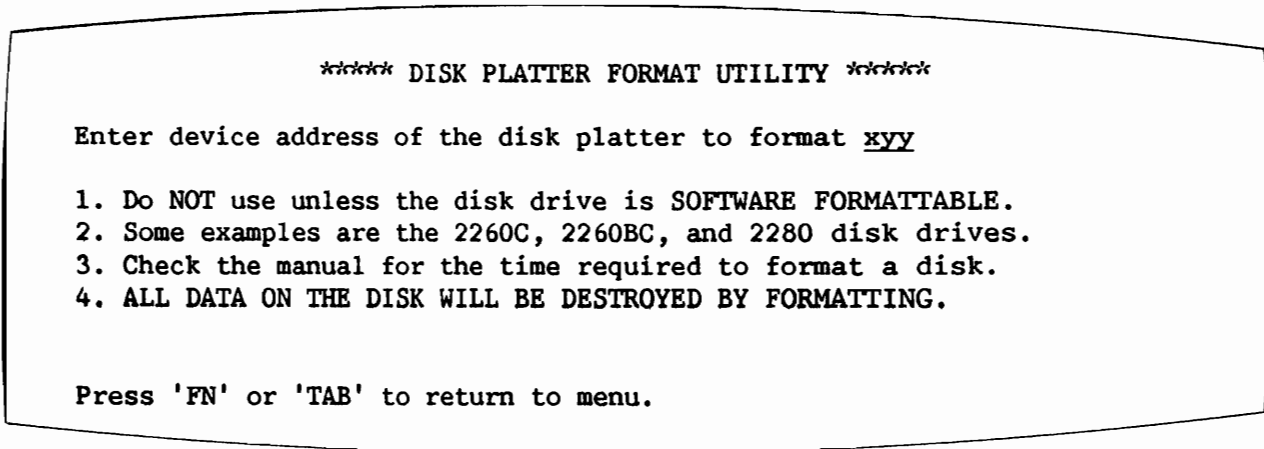


Figure 7-1. Disk Platter Format Utility Screen

Enter the device address of the disk to be formatted. When a proper device address has been entered, the CRT displays the following.

Are you sure that you want to format platter /xyy (Y/N)? \_

FORMAT PLATTER /xyy

If Y (yes) is selected, the utility continues to format the specified platter. If N (no) is selected, the utility returns to the initial display. Enter Y or N and press RETURN.

If Y is selected, the format procedure continues and the disk is checked for the presence of data. If data exists, the utility again requests if the format should be performed. If it should, then the format operation is initiated and proceeds unassisted by the operator. If an error is encountered, a message is displayed and the operator is requested to restart the utility after correcting the error condition.





CHAPTER 8  
THE VERTICAL FORMAT CONTROL UTILITY

8.1 AN OVERVIEW

Several of the printers that can be used on the 2200 Series computer are equipped with Direct Access Vertical Format Units (DAVFU). The purpose of these units is to provide the user with a means to specify customized vertical format control for printed output.

The @DAVFU utility allows the user to define and edit the vertical format data for the DAVFU. With this utility the user can create, edit, test, and save format data on disk for later loading into the DAVFU. The program may be used when vertical formats other than the printer defaults are required by programs. On printers with Forms Length Selector (FLS) controls, the user may find that using the FLS controls is more convenient and is sufficient to handle most vertical control requirements.

The operator is prompted to enter several parameters.

- . Number of lines per page
- . Number of lines per inch
- . Bottom of form

These parameters may be entered by the operator or obtained from a data file previously created by the program.

NOTE

If a data file is specified, it must be a file previously created by "@DAVFU", since the program assumes a specific format. Therefore, the first time the utility is used (since there is no data file yet), the user must use the option that allows the file to be created. The VFU format can then be stored for later usage.

After the parameters mentioned above are obtained, the user can then edit the VFU format by setting the desired Tab stops between the top and bottom of form or by deleting the unwanted tab stops. Tab stops are not allowed in Channel 1 or on Line 1, since these positions are reserved for top and bottom of form.

After the desired format is defined, the operator has the option of testing the format by specifying the number of the channel to be tested. To test the VFU format, the program skips to each tab stop on the specified channel and prints a line of text every time a new tab stop is reached. If the test is satisfactory, the option of saving the format on a data file is provided. The data file may be an already existing file, in which case it is overwritten; if the file, on the other hand, does not exist, it is created by the program. The option to edit a new VFU format is provided following all of the above.

## 8.2 RUNNING THE UTILITY

After selecting the Vertical Format Control utility from the System Utilities menu, a screen of text describing the purpose of the utility is displayed. (Refer to Figure 8-1.) Additionally, this menu describes the control codes used in DAVFU sequences.

Application programs may load and execute format files created by the @DAVFU utility. Refer to the individual printer manuals for more detailed instructions.

### DAVFU DATA PREPARATION

**Purpose:**

This program facilitates the preparation of DAVFU format data when the Forms Length Selector (FLS) is inadequate for forms control. Note, however, that FLS is more convenient for most printing.

Format data can be created, edited, and saved on disk for later loading into the DAVFU. Format data is saved in a standard 2200 data file, named by the user.

When using DAVFU sequences, the following control codes are effective:

- HEX(0B) - Vertical Tab (as specified in channel two)
- HEX(0C) - Top of form (as specified in channel one)
- HEX(0201021F0X0F) - Vertical tab (as specified in channel X)
- HEX(0201021F1X0F) - Vertical line feed (X = # of lines)

**Note:** Programs using DAVFU control sequences can load a previously defined sequence by executing the following statements in the application program:

```
DIM T$(128)
DATA LOAD DC OPEN T'filename' : REM 'filename' = name of the user file
DATA LOAD DC T$()
$GIO/xyy, (A000,G$) T$() : REM xyy represents the printer address
                          Press 'RETURN'
```

Figure 8-1. Initial DAVFU Screen



DAVFU DATA PREPARATION

No. of lines per page? 66  
 No. of lines per inch? 6  
 Bottom of form = 0

Enter 'T' for TAB stops,  
 Back space and space to delete them.  
 T in channel 1 is TOF (only 1 allowed).  
 T in channels 1 & 2 (same line) is BOF.

		CHANNEL																	
		1	2	3	4	5	6	7	8	9	10	11	12						
		-----												To move cursor use:					
L	1	+	!	T	.	.	.	.	.	.	.	.	.	.	.	!	+	S.F.'4	- LAST LINE
I	2	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-	S.F.'5	- NEXT LINE
N	3	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-	S.F.'6	- PREVIOUS LINE
E	4	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-	S.F.'7	- FIRST LINE
	5	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-	S.F.'12	- NEXT CHANNEL
	6	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-	S.F.'13	- PREVIOUS CHANNEL
	7	+	!	.	.	.	.	.	.	.	.	.	.	.	.	!	+	S.F.'20	- DONE EDITING
	8	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-	FN	- return to menu
	9	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-	TAB	- return to menu
	10	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-		
	11	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-		
	12	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-		
	13	+	!	.	.	.	.	.	.	.	.	.	.	.	.	!	+		
	14	-	!	.	.	.	.	.	.	.	.	.	.	.	.	!	-		

Figure 8-2. DAVFU Control Screen

Enter T for Tab stops; use the BACKSPACE key and the space bar to delete them. The T in Channel 1 is TOF (only 1 allowed). The T in channels 1 and 2 (same line) is BOF. These two tabs are placed automatically according to the three questions at the top of the screen in Figure 8-2.

After editing is complete, depress FN'20 to terminate editing. The system responds with the following prompt.

Is format OK? Y

If more editing is desired, enter N and press RETURN. The cursor will go to the beginning of the format. If no problems exist with the format, press RETURN. The following prompt is displayed.

Enter printer address 215

Enter the proper address and press RETURN, or press RETURN to accept the default value of 215. The system enables the user to test the format before saving the file on disk by issuing the following prompt.

VFU loaded, do you want to test format? Y

Press RETURN to test VFU, or enter N (RETURN) to proceed to next prompt and save the VFU data.

If the VFU is to be tested, enter the Channel No. to test.

As each channel is tested the printer skips to each Channel tab stop and prints a line of code.

If the test was successful, enter 0 to end channel test and the prompt requesting that the format be saved is displayed.

Do you want to save VFU? N

The format may be saved on disk for future use ('Y') or discarded ('N'). Entering 'Y' (RETURN) causes the following prompt to be displayed.

Enter Output File Name \_\_\_\_\_

Enter a file name and press RETURN.

Enter Device Address \_\_\_\_

Enter the disk address and press RETURN.

The file is saved and the prompt to edit another VFU sequence is displayed. Key N to terminate utility or Y to restart the utility.



## CHAPTER 9 DISK BACKUP

### 9.1 AN OVERVIEW

Probably the most common form of file security is file backup. File backup is simply maintaining a backup copy of important files. Backing up important files is an area that must be given high priority.

Disk storage devices are basically very reliable. However, like any other storage media, disk platters are subject to accidental damage or destruction. Losing power during an update, dropping a disk cartridge, or exposing a platter to a magnetic device are just a few of the things that could cause the destruction of data.

Most computer users cannot afford the cost and inconvenience associated with the reconstruction of a destroyed disk file. Some companies have been severely crippled when a critical file was accidentally destroyed because they did not adhere to rigid backup procedures for all essential programs and data files.

Many small computer users think the cost and time associated with maintaining backup files is high. The only cost associated with file backup is the price of an extra storage device such as a disk platter or diskette. The time involved backing up a file is minimal when the alternatives are considered.

#### When to Backup

There is no absolute rule governing the frequency of backup. It normally depends upon several factors. One important factor is the amount of activity or the number of transactions processed against a master file. With high activity, a user may wish to backup a file once a day. With fewer transactions, a frequency of once or twice a week may be sufficient.

Each user should carefully evaluate the factors relating to his/her own business and data processing requirements. As a general rule, it is recommended that important files be backed up with frequency that matches the processing activity of that file. In other words, if a file is updated daily, it should be backed up daily; if a file is updated weekly, it should be backed up weekly.

It is also a good practice to create an extra copy of a backup platter and/or keep more than one generation of these platters. Often, the backup platter can be ruined by the same problem that destroyed the original platter. Having an extra backup platter provides an additional measure of protection against time consuming and costly data-reconstruction.

Some companies also store backup copies at an off premise location. By doing so, they protect themselves against the danger of fire, explosions, or some other disaster.

## 9.2 FIXED/REMOVABLE PLATTERS OF SAME SIZE

When copying from a fixed platter to a removable platter of equal size or vice versa, a user may back up a disk in one of the four following ways.

1. COPY statement
2. MOVE statement
3. COPY/VERIFY utility
4. BACKUP utility

The BACKUP utility is primarily for use in backing up 2, 4, or 8 megabyte fixed disks onto several 1 megabyte diskettes. This utility is discussed in detail in the next section.

### COPY Statement

The COPY statement copies the entire contents of a disk platter, or a specified portion of its contents, to another disk platter in the same or different disk unit. The COPY statement is the only "Absolute Sector Addressing" mode, BASIC-2 statement that should be used in backing up a file.

Example:

```
10 COPY T/D11, (0,2000) TO T/D10,  
20 VERIFY T/D10, (0,2000)
```

Statement 10 copies sectors zero through 2000 from platter D11 to the same sectors on platter D10. Statement 20 verifies that the data recorded on the backup disk cartridge has been properly recorded.

Starting and ending sector addresses of the information to be copied should always be included in the COPY statement. If the entire contents of a disk platter is to be copied, the beginning sector address should be zero and the ending address should be the last sector on the platter. (Refer to the appropriate disk manual for this value.)

If an error is encountered following a COPY operation, the process should be repeated. Repeated failure could indicate a faulty disk platter. If the error persists with another platter, a Wang Service Representative should be called.

Additional information concerning the use of the COPY statement can be found in the Wang BASIC-2 Disk Reference Manual.



## MOVE Statement

The MOVE statement, used only with catalogued files, provides another means of backing up disk files. In addition to copying the catalogue index and catalogued files, it also provides one additional function. The MOVE statement eliminates scratched files from the catalog and compresses still-active files into the available space.

Since it only copies active files, the MOVE statement may result in a faster copy than the COPY statement. However, caution should be exercised when using MOVE that only catalogued files are on the disk platter. Any other files will be lost unless a COPY statement is used.

### Example:

```
10 MOVE T/D11, to T/D10,  
20 VERIFY T/D10,
```

Statement 10 copies all catalog information from platter D11 to platter D10. Statement 20 checks platter D10 to ensure that all information has been copied correctly.

Additional information concerning the use of the MOVE statement can be found in the Wang BASIC-2 Disk Reference Manual.

## COPY/VERIFY Utility

There is another way to back up important files. This method utilizes the ISS COPY/VERIFY utility and can be used by those customers who have purchased Wang's Integrated Support System (ISS) software package.

The COPY/VERIFY utility offers more flexibility than the COPY and MOVE statements. This utility offers the following features.

1. Copied files may be renamed and may replace existing files on the output disk
2. Selected files or all files may be copied without altering files on the output platter
3. Copying is allowed between any two platter/disk addresses.
4. The verify operation actually compares the data read from the input file to the data written on the output file to ensure that it has been copied correctly.
5. Additional sectors may be allocated for the copied file.

The operating instructions for the COPY/VERIFY utility are outlined in the Integrated Support System User Manual.

### 9.3 SYSTEMS WITH FIXED PLATTERS LARGER THAN REMOVABLE PLATTERS

The three methods of backup described in Section 9.2 only effectively work for backing up files on platters of equal size.

However, there are many cases, such as 2200LVP and 2200SVP systems, when files stored on a fixed platter of 2, 4, or 8 megabytes must be backed up onto a diskette of only 1 megabyte. In this case, the ability to copy the files to multiple output platters, thus spanning platters, is required. The @BACKUP utility is designed to provide this capability.

### 9.4 BACKUP UTILITY

The purpose of this utility is to provide the capability of copying the contents of a single disk platter to another single platter or a series of smaller platters. Both source and destination platters may be from any of the several types of disk drives used on the 2200MVP, 2200LVP, 2200VP, and 2200SVP systems. The primary use of this utility is to allow the 2-, 4-, or 8-megabyte fixed disks on the 2200LVP or 2200SVP to be backed up onto several 1 megabyte diskette platters. This utility will move the entire contents of the source platter, including the catalogue index if one is present.

On the specified output platter(s), any information presently stored will be overwritten. Additionally, since file names on the backup platter(s) are stored in a data file rather than in a copied catalogue index, the platters cannot be used for direct file access. In order to retrieve the information, the @RECOVER utility must be employed.

#### Files Created by the Utility

The following disk files are created as a result of running this utility. They are written to the output platter(s), and contain control information for purposes of recovery.

"START": Start module for the recovery program (@RECOVER). It is saved on each output platter.

"@RECOVER": Recovery program. It is saved on each output platter, if available on the input platter.

"@BADSCTR": Contains pointers to the set of bad sectors, if any, found in the portion of source disk copied onto current platter. This information is recorded in the form of a bit map.

"@INDEX": Contains a copy of the source index and is saved in every platter of the set to allow instant file recovery from any diskette.

"@DATA": Contains the portion of files copied from the source disk.

"@LABEL": Created by @BACKUP and contains the following information to be used at recovery time.

1. Backup identification key (randomly generated by the utility to uniquely identify the current set of output platters as part of the same set).
2. Date (mm/dd/yy). Date the backup is created, input by the operator.
3. Backup description.
4. Operator's name.
5. Total number of sectors of the source disk that are contained in the backup.
6. Current platter number within the set.
7. Total number of platters within the set.
8. First sector of source disk contained in current platter.
9. Last sector of source disk contained in current platter.

#### 9.5 RUNNING THE BACKUP UTILITY

After selecting the Backup Platter utility from the System Utilities menu (refer to Figure 4-1), the input address prompt with a default value of D11 is displayed. Enter the input address desired and press RETURN, or press RETURN to accept D11. The value can be permanently changed if desired by altering the A1\$ value in the program code. Since the utility is designed primarily to move files from the Winchester disk to the dual-sided, double-density diskette on an LVP or SVP system, the address D11 is selected as the default value.

If an incorrect device address is entered, the system displays an error I98, indicating that the platter requested is not mounted. Press RETURN to go back to the initial prompt.

Following the successful selection of the input address, the prompt for an output address is displayed. The default value of D10 is displayed. Enter the address of the output (backup) diskette and press RETURN, or press RETURN to accept D10. The next prompt displayed is for the date.

Today's date is: mm/dd/yy

Enter the date in the indicated format and press RETURN. This date will be saved on the backup disk in the @LABEL file.

A prompt for a backup description is then displayed. Enter a text description of up to 60 characters in length and press RETURN. This is also stored in @LABEL.

Next, enter the name of the operator. A maximum of 50 alphanumeric characters is allowed; press RETURN.

The system now displays the starting and ending sectors for the copy. The starting sector will be 0 so that the catalogue, if it exists, is also copied and the default value for the end is the current end of catalogued data. The end of the catalogue area is also displayed. For example:

Copy starts at sector 0 and ends at sector: 43828

Options:

Current end (End of Catalogue Data) - 43828

End of Catalogue Area (As specified in the Index) - 52607

Enter the absolute address of the last sector to copy and press RETURN, or press RETURN to accept the current end of catalogued data.

The system then verifies the input index and checks for valid data on the output platter. The message at the bottom of the screen is displayed if the output contains valid data. (Refer to Figure 9-1.)

Press RETURN and mount a new platter at the output address. Enter Y and press RETURN. After mounting the new platter, or if the output platter did not contain valid data, the following prompt is displayed.

Number of Sectors in Output Platter 01232

The default value is the end of catalogue area as specified in the index. Enter the absolute address of last physical sector in the output disk and press RETURN, or press RETURN to accept the default value.

The system now scratches the output platter, creates the file 'START', and copies the '@RECOVER' utility on the output disk. It then displays a message indicating the number of platters needed to copy the information, which platter is currently being copied to, and how many sectors are contained there. The unique Backup ID# is also displayed near the top of the CRT screen. This information should be copied onto the output disk's label for future reference when recovering the data.

On the bottom of the screen, messages are displayed indicating that the data files described are being created on the output platter "START", "@RECOVER", "@LABEL", "@BADSCTR", and "@INDEX".

Across the bottom of the screen, a box is displayed indicating the progress of the copy. As the box is filled, the file is being copied. The input is first verified, the copy is performed, and then the output is verified.

As additional platters are needed, a prompt to mount the next diskette in sequence is displayed. Mount the requested platter and press RETURN. The messages will again be displayed indicating that the copy is in progress. When the backup has been completed, an END OF BACKUP message is displayed.

```
          **** BACKUP UTILITY ****

Input Address:      D11           Output Address:      D10

Today's Date is:   05/19/81

Backup Description: Backup Test

Operator's Name:   Author's Name

Copy starts at sector 0 and ends at sector: 14176

Warning: The contents of the output platter will be replaced by the backup
data. Your output platter seems to contain valid data.
Are you sure you want to use this platter? (If not, mount a new platter) N
```

Figure 9-1. Backup Utility Screen

9.6 RECOVER UTILITY

@RECOVER is the complementary utility to @BACKUP; it retrieves the contents of the platters produced by @BACKUP and writes them onto another platter. There are three options provided by the @RECOVER utility for recovering data.

1. Recover the entire contents of a backup platter or platters. In cases where an exact copy of the original is required or when the source was an uncatalogued platter, this option must be used.
2. Recover all active files on the backup platter or platters. If there are scratched files on the backup platter(s), this provides the means for eliminating them.
3. Recover selected files from the backup platter(s). With this option, individual files can be recovered from the backup. The recovered files can be added to a catalogued disk which does not have to be identical to the source platter. Additionally, this option enables the file size to be enlarged.

The Recover utility must only be used to recover files from disk platters created by the @BACKUP utility. For general file move capability the @MOVEFIL utility should be used.

## 9.7 RUNNING THE RECOVER UTILITY

After selecting the Recover from Backup utility from the System Utilities menu (refer to Figure 4-1), the prompts for the input and output addresses are displayed. The addresses D10 and D11 are set up as defaults for input and output respectively. They are reversed from the defaults set in the @BACKUP utility because generally @RECOVER is performed on diskettes created by @BACKUP and recovered to hard disk.

Enter the input address and press RETURN, or press RETURN to accept the default value of D10.

Enter the output address and press RETURN, or press RETURN to accept the default value of D11.

The input platter is validated, and the Recovery Utility Screen, as shown in Figure 9-2, is displayed. The information displayed on this screen is obtained from the @LABEL file created by the @BACKUP utility. The date of backup, backup identification #, description, and author's name are all obtained during @BACKUP and stored for use by @RECOVER in @LABEL.

```
***** RECOVERY UTILITY *****

Input Address: D10                      Output Address: D11

Date of Backup: 12/02/80                Backup Identification #: 853758

Description: Backup Test

Created by: Author's Name

This is platter 1 of 2 which contains sectors 0 - 1215
of original disk.

Enter the number of the desired option: __
1-Recover entire disk (WARNING! Output disk will be scratched)
2-Recover all active files (WARNING! Output disk will be scratched)
3-Recover specified files

Press 'FN' or 'TAB' to return to menu.
```

Figure 9-2. Recovery Utility Screen

NOTE

Recovery requires that backup platters be processed in order and contain the same ID# key. Discrepancies are displayed.

### Recovering Specific Files

If option 3, "Recover specified files" is selected, a prompt for a printout of files that can be recovered is displayed.

A printout is provided listing the files on the set of backup platters. This list corresponds to that produced by the LIST DC statement for the original source platter. Selected files can be input for recovery.

A prompt is displayed requesting the name of the file to be recovered. Enter the name and press RETURN. The file is now recovered to the output platter. If the Backup set contains more than one diskette, the operator is prompted to insert each diskette in the set in sequence, with the same Backup identification number. If the file to be copied already exists on the output platter, an overwrite feature is provided.

When the selected file has been recovered, the prompt requesting another file name is displayed again. Enter the name and press RETURN, or press RETURN to end the recover function.

If @BADSECT indicates that a bad sector was found by @BACKUP in a file on the original source disk, the operator is informed. @RECOVER still moves the file to the output disk, but the bad sector is filled with Hex(00) codes. About the only action that can be taken at this point would be to scratch the file on the output disk and attempt to recover it from a previous backup.

### Recovering All Active Files

To recover only active files, select Option 2. The following prompts are displayed.

Do you want to change the index size on the output platter? N

Default output index size (same as in backup data): 1

The @INDEX file is read to obtain the index parameters of the original source platter. At this time, the operator can alter the size of the index. Type Y (RETURN) to change index size and then provide the system with the desired index size, or press RETURN to accept current size.

Mount input platter that is first in the series when requested and press RETURN. The system now recovers all active files to the output platter.

If the output platter does not contain enough file space to recover the input platter, an error message is displayed. Press RETURN to discontinue processing.

### Recovering Entire Disk

To recover the entire disk, select Option 1. The system proceeds to recover the entire input disk. The percentage of the disk recovered is displayed on the bottom of the screen.

The @INDEX file is read to obtain the index parameters of the original index before the destination platter is scratched. The destination platter is then scratched. Each active file in @INDEX is located and moved to the destination platter.

If @BADSECT indicates the file contains bad sectors, the operator is alerted. @RECOVER still moves the file to the output disk, but the bad sector is filled with Hex(00) codes. About the only action that can be taken at this point would be to scratch the file on the output disk and attempt to recover it from a previous backup.

As each input platter is recovered, the operator is prompted to insert the next diskette in sequence. When recovery is complete, and END RECOVERY message is displayed.

### Error Messages

<u>MESSAGE</u>	<u>ACTION</u>
This platter does not belong to the current set.	Change input platter.
Error in sector #___. Program will be terminated. Output disk must be reformatted.	Press RETURN. Begin procedure again after output disk has been formatted.
File size exceeds available disk space.	Press RETURN.
Input platter illegal. (Not a platter created by @BACKUP.	Mount the correct input platter and press RETURN.

If bad sectors are encountered, the operator may obtain a listing of these bad sectors.



CHAPTER 10  
TRANSPORTING SOFTWARE

10.1 AN OVERVIEW

The introduction of the 2200LVP and 2200SVP systems has generated the need for a new method of transporting software. There are two reasons for this need.

1. The diskette (1MB) and the fixed disk platters (2-8MB) differ in size. Thus, the BASIC-2 MOVE statement cannot be used to move files larger than 1MB from fixed disk to diskette. Furthermore, the Integrated Support System (ISS) does not support multi-volume file move. Moving large files onto diskettes thus becomes difficult.
2. The 2200LVP and 2200SVP do not support the standard Wang SSSD format diskettes. Moving software between a 2200VP or 2200MVP and a 2200LVP or 2200SVP is not possible with any previous means of software transportation.

The Move File utility (@MOVEFIL) provides the user with a general file move capability. Selected files or all active files are moved from any platter to any other. The utility provides the following capabilities.

1. Files can span more than one output platter; this facilitates moving from fixed disk to a set of diskettes.
2. New files can be created with specified free space.
3. The size of the new output file can be changed.
4. Files can be renamed.
5. Existing files can be overwritten.
6. All active files can be moved with options to overwrite all files or not overwrite any files.

10.2 THE MEDIA

On the standard 2200 Format Platter (i.e., with 256-byte sectors) files are recorded using the standard Wang 2200 disk catalogue structure. The output platter must be initially formatted and scratched before using the @MOVEFIL utility.

For software interchange between the 2200VP and 2200MVP systems with 2270A diskette drives and the 2200LVP and 2200SVP systems with Wang DSDD diskettes, pre-formatted 3741-type platters must be utilized. These platters may be distinguished from other Wang-supplied diskettes by their green labels. Before they may be used by the @MOVEFIL utility, the 3741 diskettes must be formatted on the 2200LVP or 2200SVP diskette drive. The Format Disk Platter (@FORMAT) utility may be used for this purpose.

### 10.3 @MOVEFIL 3741 FORMAT

The first track of the platter is used as the index, for a total of 26 index sectors (128 bytes each). Sector 0 is initialized to show the following information.

Index sectors = 26 (Since each sector is only 128 bytes long, the equivalent in 2200 format is 13 sectors.)

Current end = 25

End catalogue area = 18385 (last available sector in a 3741 platter).

Files are allocated sequentially (no hashing is used) and the index entries are identical to those used in standard 2200 catalogue structure.

### 10.4 @MOVEFIL RESTRICTIONS

1. Written in BASIC-2, this utility will not run on older models of the 2200, more specifically the 2200A, B, C, S, or T systems.
2. The utility does not format or scratch any platters.
3. The move operation will terminate if a bad sector is encountered on either the input or output platter.
4. The utility requires a 13K partition for execution.

### 10.5 MULTI-VOLUME FILES

#### Output Files

If the file to be moved is too large for the output platter, a file named @SPAN001 is created. This file occupies the remainder of the space on the output platter. The @SPAN001 file contains the following information.

- . the actual name of the output file
- . the actual size of the output file
- . as much of the file as will fit

An additional output platter must then be mounted. A file named @SPAN002 is created on this platter. If necessary, a third platter with @SPAN003 is created and so forth.

## Input Files

If the file to be moved is not in the platter index, the utility searches for @SPAN001. If @SPAN001 contains the file to be moved, it then determines if the file spans from this platter to any subsequent platters. If so, the file is moved and the utility prompts the operator for the additional platters to be mounted. The digits in the @SPAN name and the name of the actual file are verified each time a new input platter is mounted.

## Multi-volume File Format

A multi-volume file will always be a data file. The first sector will contain the following.

- Byte 1: Status (HEX(10) if active file; HEX(11) if data file).
- Byte 2: Type (HEX(80) if program file; HEX(00) if data file).
- Bytes 3-4: The size of the original input file.
- Bytes 5-6: Number of sectors used by the original input file.
- Bytes 7-8: Portion of input file contained on the current platter (i.e., binary equivalent of last three digits in "SPANxxx").
- Bytes 9-17: Name of the original file.

All of this information is recorded using absolute sector addressing (i.e., DATALOAD BA....). The remainder of the file contains a portion of the original file and a standard 2200 file trailer.

## 10.6 RUNNING THE UTILITY

After selecting the Move File utility from the System Utilities menu (refer to Figure 4-1), the prompts requesting input and output addresses, as well as the platter type, are displayed.

Enter the input address and press RETURN, or press RETURN to accept the default value. Enter the platter type, either W (Wang) or I (IBM), and press RETURN; or press RETURN to accept the default value. Repeat this procedure for the output file.

After the input address, output address, and platter type are entered, the screen in Figure 10-1 is displayed.

Enter Y or N and press RETURN. If Y (yes) is selected, the utility automatically displays the following.

\*\* Moving All Active Files \*\*

Do you wish to overwrite files? Y

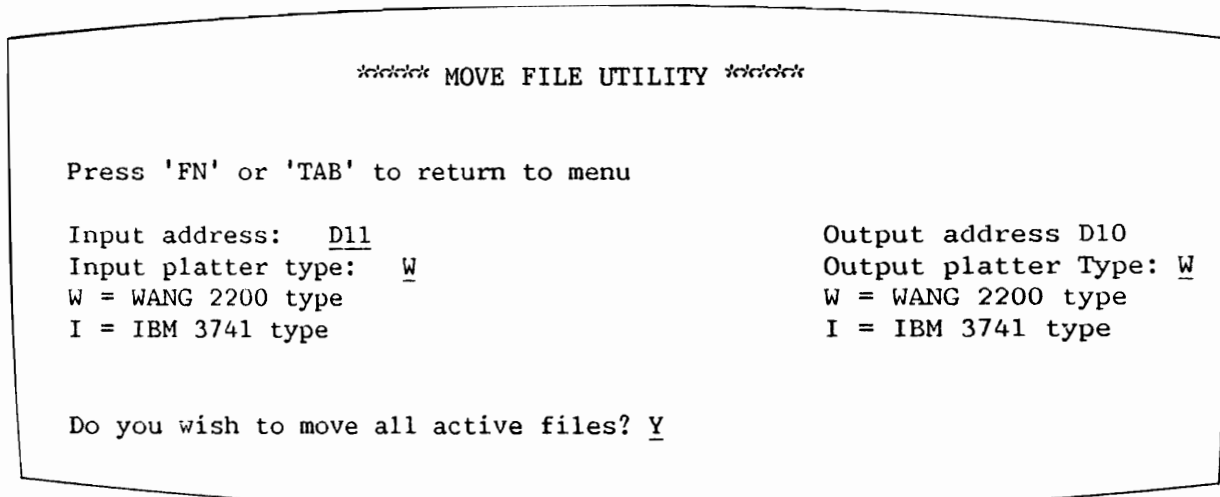


Figure 10-1. Move File Utility Screen

If Y is selected, all files of the same name on the output disk will be automatically overwritten. If N is selected, the utility begins to copy. If a duplicate file is found, the program ignores that file and continues on to the next file. The output index is scanned and parameters are obtained. All files that are not duplicates are automatically copied to the output platter. Files that already exist on the output platter and were not overwritten are then displayed on the screen as follows.

OVERWRITE EXCEPTIONS

1) Filename 1	11)	21)
2) Filename 2	12)	22)
.	.	.
.	.	.
.	.	.
10)	20)	30)

Since these files have not been moved, the user must return to the options menu and re-run the Move File utility selecting to overwrite these files on an individual basis.

If the user does not wish to automatically move all active files, each file name from the input disk is displayed along with its parameters, beginning with the first file name in the index. As the utility steps through each file, if the user does not want that file moved, then entering blanks will ignore this file and pick up the next file in the index. If the user changes the Input file name at any time (other than blanks) the new file name becomes the default input name until it is changed again. The CRT displays the following prompts.

Input file name: FILEOUT

Extra Sectors: 0004

Currently, 6 sectors are used in FILEOUT and 4 are free.

Enter the number of extra sectors to be given to the output file and press RETURN. The following prompt is now displayed.

Output file name: FILEOUT

The system automatically assumes the same name for the output file. If the file name is to be different, enter the new name and press RETURN. The file will be automatically moved to the output platter. If the name already exists on the output platter, a prompt to overwrite this file is displayed.

File FILEOUT already exists. OK to overwrite? N

If it is not necessary to overwrite the file, press N and the cursor returns to the output file name prompt. Enter a new file name and press RETURN. The file is then moved to the output platter with a new file name. If Y is typed, the file is moved, overwriting the existing file on the output platter.

If the output media is a 3741 platter, the user is given the option of initializing the diskette to @MOVEFIL 3741 format.

Do you want to initialize the platter? Y

Press RETURN if you would like the platter initialized, or press N (RETURN) if you don't want the platter initialized.

Output file name \_\_\_\_\_

The name of the input file will be defaulted. Press RETURN to accept the default name, or enter a new name. Names of the form "@SPANxxx" are rejected. When @SPAN001 is found, @MOVEFIL displays the actual file name that is spanned; in this way the user knows the input file name and can rename it if desired.

A message is displayed indicating a scan of the output index is being made. The utility checks to see if the file already exists. If so, the following is displayed.

File "name" already exists. OK to overwrite? N

Press Y (RETURN) to copy, or simply press RETURN to return to the file name prompt.

Output parameters are calculated and the file is copied to the output platter. The next file in the input index is obtained and the procedure will be repeated with that file name as a default.



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.

**700-6855**

TITLE OF MANUAL **BASIC-2 Utilities Reference Manual**

COMMENTS:

\_\_\_\_\_  
Fold

\_\_\_\_\_  
Fold



Fold



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY CARD**  
FIRST CLASS      PERMIT NO. 16      LOWELL, MA

POSTAGE WILL BE PAID BY ADDRESSEE

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

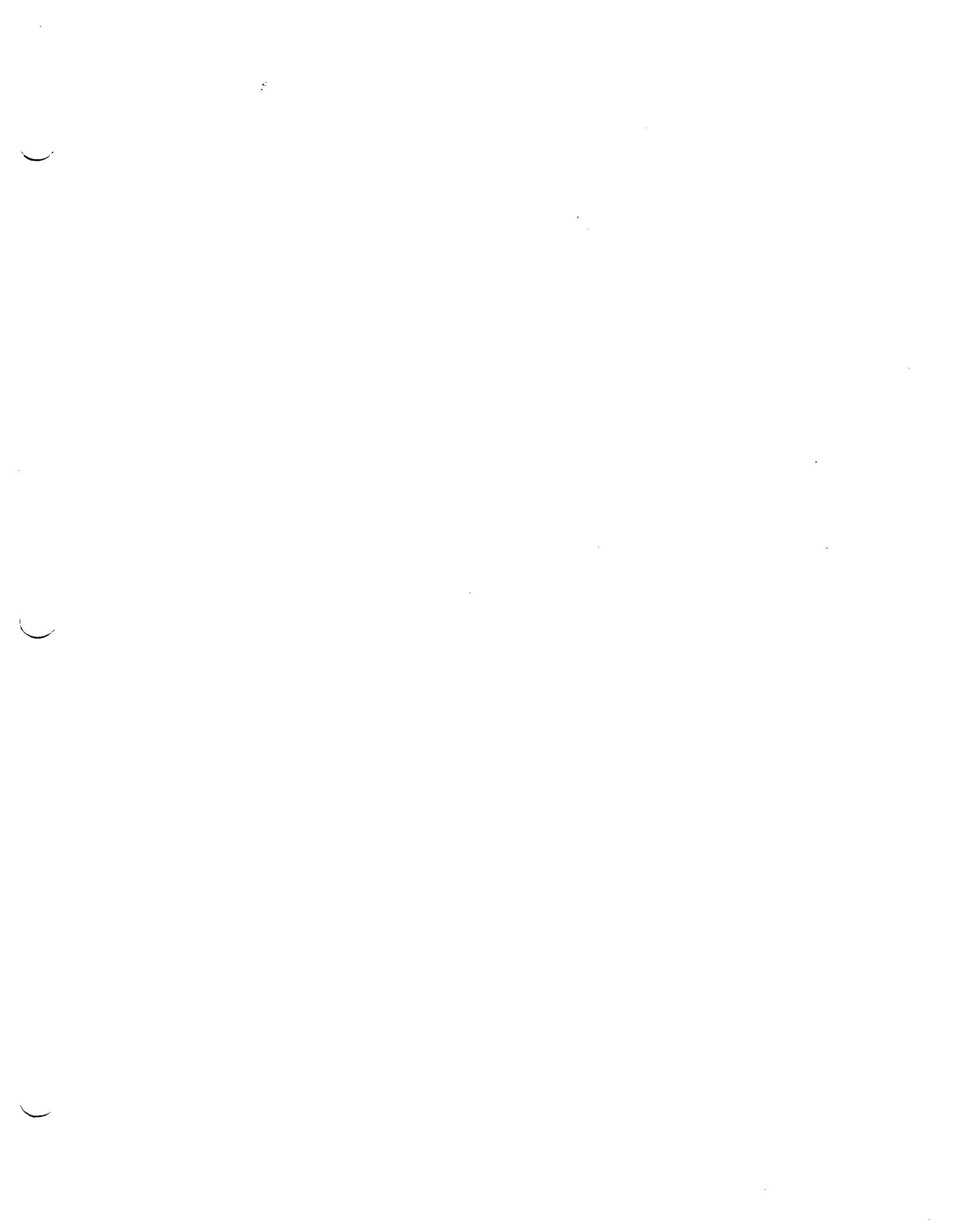


Cut along dotted line.

Attention: Technical Writing Department

Fold





## International Representatives

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

## United States

<b>Alabama</b> Birmingham Mobile	<b>Florida</b> Coral Gables Hialeah Hollywood Jacksonville Miami Orlando Sarasota Tampa	<b>Iowa</b> Ankeny <b>Kansas</b> Overland Park Wichita	Southfield <b>Minnesota</b> Eden Prairie Minneapolis	Syossett Syracuse Tonawanda	<b>South Carolina</b> Charleston Columbia
<b>Alaska</b> Anchorage Juneau	<b>Georgia</b> Atlanta Savannah	<b>Kentucky</b> Louisville	<b>Mississippi</b> Jackson	<b>North Carolina</b> Charlotte Greensboro Raleigh	<b>Tennessee</b> Chattanooga Knoxville Memphis Nashville
<b>Arizona</b> Phoenix Tucson	<b>Hawaii</b> Honolulu Maui	<b>Louisiana</b> Baton Rouge Metairie	<b>Missouri</b> Creve Coeur St. Louis	<b>Ohio</b> Akron Cincinnati Cleveland Independence Toledo Worthington	<b>Texas</b> Austin Dallas El Paso Houston San Antonio
<b>California</b> Anaheim Burlingame Culver City Emeryville Fountain Valley Fresno Los Angeles Sacramento San Diego San Francisco Santa Clara Ventura	<b>Idaho</b> Boise	<b>Maine</b> Portland	<b>Nebraska</b> Omaha	<b>Oklahoma</b> Oklahoma City Tulsa	<b>Utah</b> Salt Lake City
<b>Colorado</b> Englewood	<b>Illinois</b> Arlington Heights Chicago Morton Oakbrook Park Ridge Rock Island Rosemont Springfield	<b>Maryland</b> Baltimore Bethesda Gaithersburg Rockville	<b>Nevada</b> Las Vegas	<b>Virginia</b> Newport News Norfolk Richmond Rosslyn Springfield	<b>Washington</b> Richland Seattle Spokane
<b>Connecticut</b> New Haven Stamford Wethersfield	<b>Indiana</b> Fort Wayne Indianapolis South Bend	<b>Massachusetts</b> Boston Burlington Chelmsford Lawrence Littleton Lowell Methuen Tewksbury Worcester	<b>New Hampshire</b> Manchester	<b>Wisconsin</b> Appleton Brookfield Green Bay Madison Wauwatosa	
<b>District of Columbia</b> Washington		<b>Michigan</b> Grand Rapids Kalamazoo Lansing	<b>New Jersey</b> Bloomfield Clifton Edison Mountainside Toms River		
			<b>New Mexico</b> Albuquerque Santa Fe	<b>Oregon</b> Eugene Portland Salem	
			<b>New York</b> Albany Jericho Lake Success New York City Rochester	<b>Pennsylvania</b> Allentown Erie Harrisburg Philadelphia Pittsburgh State College Wayne	
				<b>Rhode Island</b> Providence	

## International Offices

<b>Australia</b> Wang Computer Pty., Ltd. Adelaide, S.A. Brisbane, Qld. Canberra, A.C.T. Perth, W.A. South Melbourne, Vic 3 Sydney, NSW	Victoria, B.C. Winnipeg, Manitoba	<b>Japan</b> Wang Computer Ltd. Tokyo	Malmö Switzerland Wang A.G. Zurich Basel Bern Geneva Lausanne St. Gallen
<b>Austria</b> Wang Gesellschaft, m.b.H. Vienna	<b>China</b> Wang Industrial Co., Ltd. Taipei	<b>Netherlands</b> Wang Nederland B.V. IJsselstein Groningen	<b>Wang Trading A.G.</b> Zug
<b>Belgium</b> Wang Europe, S.A. Brussels Erpe-Mere	<b>France</b> Wang France S.A.R.L. Paris Bordeaux Lille Lyon Marseilles Nantes Nice Rouen Strasbourg	<b>New Zealand</b> Wang Computer Ltd. Auckland Christchurch Wellington	<b>West Germany</b> Wang Deutschland, GmbH Frankfurt Berlin Cologne Düsseldorf Essen Freiburg Hamburg Hannover Kassel Mannheim Munich Nürnberg Saarbrücken Stuttgart
<b>Canada</b> Wang Canada Ltd. Burlington, Ontario Burnaby, B.C. Calgary, Alberta Don Mills, Ontario Edmonton, Alberta Halifax, Nova Scotia Hamilton, Ontario Montreal, Quebec Ottawa, Ontario Quebec City, Quebec Toronto, Ontario	<b>Great Britain</b> Wang (U.K.) Ltd. Richmond Birmingham London Manchester	<b>Panama</b> Wang de Panama (CPEC) S.A. Panama City	
	<b>Hong Kong</b> Wang Pacific Ltd. Hong Kong	<b>Puerto Rico</b> Wang Computadoras, Inc. Hato Rey	
		<b>Singapore</b> Wang Computer (Pte) Ltd. Singapore	
		<b>Sweden</b> Wang Skandinaviska AB Stockholm Gothenburg	

# WANG

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

Printed in U.S.A.  
700-6855  
3-82-2M