WANG LABORATORIES, INC.

Computers

MARKETING RELEASE

TO DISTRIBUTION	PUBLICATION ≠
FROM MARKET PLANNING AND DEVELOPMENT	DATE AUGUST 1980
SUBJECT MVP/LVP OPERATING SYSTEM RELEASE 2.1	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION □ YES □ NO

Wang Laboratories, Inc. announces version 2.1 of the multi-user MVP Operating System. This release provides several new features:

FEATURES

- Supports both 2200MVP and 2200LVP.
- Several new utilities are provided including Backup, Recovery, Move.

RESTRICTIONS

- All 2200MVP systems should be upgraded to Release 2.1.
- All 2200LVP systems require Release 2.1.

ORDERING INFORMATION

Order through Software Distribution:

	2200MVP	2200LVP
Package Number	195-0049-3	195-2162-5
Diskette Number	701-2294M	704-0002B

AVAILABILITY

In stock, allow one week for delivery.

PRICE

No charge to 2200MVP, 2200LVP users.

SUPPORT

This is a Category 1, Wang-supported product. Any suspected errors or anomalies found in this package should be documented and forwarded to Wang Laboratories via the local district analyst.

MVP BASIC-2 RELEASE 2.1

This release is available on the following media:

1. Part # 701-2294M: SSSD Diskette (2270A).
2. Part # 704-0002B: DSDD Diskette (2200 LVP).

Release 2.1 of MVP BASIC-2 replaces all previous MVP BASIC-2 releases. This release provides all 2200MVP and 2200LVP systems with several new features and corrects all known system anomalies. The system platter includes the MVP (multi-programming) Operating System and BASIC-2 language processor, system diagnostics, and several utilities. Release 2.0 had a limited distribution to initial 2200LVP deliveries only.

I. SYSTEM UTILITIES

The utilities described below can be accessed by entering LOAD RUN (RETURN). A menu will be loaded providing access to the utilities. Certain utilities are for particular devices and do not have a function in all 2200 configurations.

@GENPART: Partition Generation

This utility creates, saves, and executes system configurations which divide the 2200 resources among the system users. (See 2200MVP Introductory Manual or 2200LVP Introductory Manual).

@PSTAT: Partition Status

This program displays the current status of each partition in the current configuration. (See $PSTAT in the BASIC-2 Language Reference Manual).

@MOVE: Move System Files

This utility moves specified system files from one disk platter to another.

@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. (See program REMarks for customization). The system platter contains a START program that merely overlays in @MENU.

@FORMAT: Format Disk Platter

This program formats software formattable disk platters, such as 2260C, 2260BC, 2280 platters, dual sided double density diskettes and SVP fixed platters. (Refer to the appropriate disk reference manual for detailed formatting information).

@2273VFU: 2273 Vertical Format Control

This utility defines 2273 Band Printer vertical format control sequences. (See 2273 Reference Manual.)

@BACKUP: Platter Backup

This program provides a multi-volume platter backup capability. It is particularly useful for systems, such as 2200LVP's or 2200SVP's, which have different size fixed and removable platters.

@RECOVER: Backup Recovery

This is the companion recovery program for @BACKUP. The entire platter, active files only, or selected files can be recovered from the backup platter(s).

@MOVEFIL: Move File

This program moves selected files from one platter to another. If fields are too large, multiple platters will be used. File data can be recorded in 3741 format for software transport between VP/MVP and SVP/LVP systems.

II. SYSTEM CHANGES

Listed below are the system changes since Release 1.9. The following files on the system platter differ from Release 1.9:

@@	-	MVP OS & BASIC-2 language processor
@MENU	-	Menu Utility
.SYSMVPB	-	Menu Mode Data
@2273VFU	-	2273 Vertical Format Control
@FORMAT	-	Format Disk Platter
@MOVE	-	Move System Files
@PSTAT	-	Partition Status Utility
@BACKUP	-	Platter Backup (new)
@RECOVER	-	Backup Recovery (new)
@MOVEFIL	-	Move File (new)
@MOVE1	-	Move File overlay (new)

A. SYSTEM ENHANCEMENTS

1. MVP BASIC-2 now supports the 2200LVP as well as the 2200MVP hardware.

2. The following new utilities are provided: @BACKUP, @RECOVER, and @MOVEFIL.

B. CORRECTED ANOMALIES

1. Attempting to execute the MAX function of an alpha array did not generate an error.

2. A system crash could occur due to the illegal use of trailing minus signs in the PACK statement image if other programs were running in other partitions. The system inadvertently allowed the illegal syntax.

3. The Partition Status Utility, @PSTAT, could occasionally erroneously display the bank separation lines.

4. The Format Disk Platter Utility, @FORMAT, did not release ($CLOSE) the disk after formatting.

5. The 2273 Vertical Format Control utility, @2273VFU, did not treat bottom of form properly. The BOF was specified at the last printed line; but, in fact, no data can be printed at the BOF line. BOF is now specified after the last line to be printed and can be omitted if there is not to be a bottom of form skipover.

6. If RESET were pressed while a disk operation was in progress for that partition, ERR90 could result on a second partition that was waiting to access that disk.

@BACKUP AND @RECOVER UTILITY DOCUMENTATION

GENERAL INFORMATION

The backup utility, @BACKUP, provides the ability to copy the contents of a single disk platter to another platter or to a series of smaller platters. The source and destination platter may be from any of the several types of disk drives used on the model 2200VP, 2200MVP, 2200SVP, or 2200LVP computers. The primary purpose of @BACKUP is to allow the 2, 4, or 8 megabyte fixed disks on the 2200LVP to be backed up onto several 1 megabyte floppy platters. This utility moves the entire contents of the source platter including the catalogue index if one is present. The complementary utility, @RECOVER, enables information produced by @BACKUP to be recovered onto another platter. Three options are provided for recovering data:

1. Recover the entire content of the backup platter(s).
2. Recover all the active files on the backup platter(s).
3. Recover only selected files from the backup platter(s).

Recovering the entire disk. If an exact copy of the original source platter is required, or if the source was an uncatalogued platter, this option must be used.

Recovering all active files. This option provides an efficient means of removing all scratched files from the backup platter(s).

Recovering selected files. Individual files may be recovered from the backup platter(s) using this option. The files that are recovered may be added to a catalogued disk which need not be identical to the original source platter. This option also provides the ability to enlarge the size of the file if desired.

OPERATING INSTRUCTIONS - @BACKUP

All of the following parameters are provided by the operator, in the order in which they are mentioned, and stored in a data file called @LABEL for future use. This data file is discussed under "Disk Files".

1. Input disk address (default = D11).

2. Output disk address (default = D10).

3. Date (i.e., mm/dd/yy).

4. Backup Description.

5. Operator's name.

6. Absolute address of last sector to copy (default is the current end of disk as specified in the index, but copying to any existent sector is permitted).

7. Absolute address of last physical sector in output disk (default is the end of catalogue area as specified in the index, if this infomation proves meaningful).

DISK FILES

The following data files are created on the backup platter(s), and contain control information for recovery purposes.

1. "START": Start module for the recovery program. It is saved in each output platter.

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

3. "@LABEL": Contains the following information to be used at recovery time:

   a. Backup identification key (randomly generated to uniquely identify the current set of output platters as part of the same set.
   b. Date (mm/dd/yy).
   c. Backup description.
   d. Operator's name.
   e. Total number of sectors of the source disk that are contained in the backup.
   f. Current platter within set.
   g. Total number of platters in set.
   h. First sector of source disk contained in current platter.
   i. Last sector of source disk contained in current platter.

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

5. "@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.

6. "@DATA": Contains the portion of files copied from source disk into the current diskette. If any bad sectors were found in this part of the input disk, they are replaced in the output platter with null sectors (i.e., HEX(00)'s are saved in the corresponding sector in the output disk).

OPERATING INSTRUCTIONS - @RECOVER

Recovering the entire contents of the original source platter.

1. The operator must supply the address of the input disk (i.e., the platter(s) produced by @BACKUP), and the address of the output platter.

2. The operator must also supply the date (mm/dd/yy).

3. The destination platter is scratched.

4. The contents of the @LABEL file are displayed on the terminal. NOTE: @RECOVER requres that backup platters must be processed in order and that each platter must contain the same ramdom identification key. Any discrepency is reported to the operator.

5. The contents of @DATA are copied to the destination platter. The appropriate sectors on the destination platter are verified, and the recovery process is halted if bad sectors are found.

6. Steps 4-5 are repeated for each backup platter in the set.

RECOVERING ALL ACTIVE FILES

1. The operator must supply the address of the input disk (i.e., the platter(s) produced by (@BACKUP), and the address of the output platter.

2. The operator must supply the date (mm/dd/yy).

3. The contents of the @LABEL file are displayed on the terminal. NOTE: @RECOVER requires that backup platter(s) must be processed in order and that each platter must contain the same random identification key. Any discrepency is reported to the operator.

4. The destination platter is scratched. The @INDEX file is read to obtain the index parameters of the original source platter. The operator is allowed to alter the size of the original index before the destination platter is scratched.

5. @RECOVER locates the next active file in @INDEX and moves it to the destination platter. If @BADSCTR Indicates the file contains bad sectors, the operator is alerted.

6. Step 5 is repeated until all backup platters have been processed.

RECOVERING SPECIFIED FILES

1. The operator must supply the address of the input disk (i.e., the platter(s) produced by @BACKUP) and the address of the output platter.

2. The operator must supply the date (mm/dd/yy).

3. The contents of @LABEL are displayed on the terminal.

4. The operator must supply the name of the file to be recovered. The size of this file may be increased if desired.

5. @RECOVER verifies that the file is on the current backup platter; if not, the operator is requested to mount the correct platter.

6. If the file currently exists on the destination platter; the operator is given the option of cancelling recovery or overwriting the file.

7. If @BADSCTR indicates that the file contains any bad sectors, the operator is informed of this fact.

8. Steps 4-8 are repeated as long as the operator wishes.
NOTE: When using this option, the operator may request @RECOVER to produce a hard copy printout showing the list of active files in the set of backup platters.

@2273VFU Program Documentation

The '@2273VFU' program is a utility that allows the user to edit the vertical format unit on the 2273 Band Printer by direct memory access. This program may be used when running applications that require special print formats not supplied by the VFU defaults.

Program Description

The Input parameters are: Number of lines per page, number of lines per inch and bottom of form. These parameters can be either supplied by the operator or retrieved from a data file previously created by the program. Note: If a data file is specified, it must be a file previously created by "@2273VFU", since the program assumes a specific format (therefore, the very first time that the program is used, since there is not a data file yet, the user must use the option that allows him to create the file, so that the VFU format can be stored for later usage).

After the input is obtained, the operator is allowed to edit the VFU format by setting the desired Tab stops between the top and the 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.

When 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 that 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 will be overwritten; if the file, on the other hand, does not exist, it will be created by the program.

After all of the above has taken place, the option of editing a new VFU format is provided.

The operating instructions are clearly documented on the CRT, and appropriate error checking and data validation are included. To start running the program, the address of the MVP system diskette must be selected. Then, the operator must type 'LOAD RUN "@2273VFU", and an explanatory display of the program function will appear on the screen, followed by the message 'Key Return'. Upon proper response to the prompt, the program will then allow the operator to provide the input parameters and edit the VFU format.

Note: For further information regarding usage of DAVFU, refer to 'Model 2273 Band Printer User Manual', Section 4.4.

OPERATOR INSTRUCTIONS FOR THE 2273 VERTICAL FORMAT CONTROL UTILITY

To start the program, you must select the address of the MVP system diskette. Next, type 'LOAD RUN "@2273VFU"'. Upon proper response to the prompt, the system is ready to accept input parameters and editing of VFU formats. LOAD RUN "@2273VFU" causes the following text to appear on the screen:

Purpose:

This program facilitates the preparation of 2273 DAVFU format data when the FLS (Forms Length Selector) is inadequate for forms control. Note, however, that the FLS (Forms Length Selector) 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(OB)   - Vertical Tab (as specified in Channel 2)
    HEX(OC)   - Top of Form (as specified in Channel 1)
    HEX(1FOX) - Vertical Tab (as specified in Channel X)
```

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

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

Press 'RETURN'

RETURN causes the following to be displayed on the screen:

```
No. of lines per page?                  Enter 'T' for TAB stops,
No. of lines per inch?                  Back space and space to delete them.
Bottom of form is at line?              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:
            -------------------------------------    S.F.'4  - LAST LINE
L   1      +!T  .  .  .  .  .  .  .  .  .  .  .!+    S.F.'5  - NEXT LINE
I   2      -!.  .  .  .  .  .  .  .  .  .  .  .!-    S.F.'6  - PREVIOUS LINE
N   3      -!.  .  .  .  .  .  .  .  .  .  .  .!-    S.F.'7  - FIRST LINE
E   4      -!.  .  .  .  .  .  .  .  .  .  .  .!-    S.F.'12 - NEXT CHANNEL
    5      -!.  .  .  .  .  .  .  .  .  .  .  .!-    S.F.'13 - PREVIOUS CHANNEL
    6      -!.  .  .  .  .  .  .  .  .  .  .  .!-    S.F.'20 - DONE EDITING
    7      +!.  .  .  .  .  .  .  .  .  .  .  .!+         FN - RESTART
    8      -!.  .  .  .  .  .  .  .  .  .  .  .!-
    9      -!.  .  .  .  .  .  .  .  .  .  .  .!-
    10     -!.  .  .  .  .  .  .  .  .  .  .  .!-
    11     -!.  .  .  .  .  .  .  .  .  .  .  .!-
    12     -!.  .  .  .  .  .  .  .  .  .  .  .!-
    13     +!.  .  .  .  .  .  .  .  .  .  .  .!+
    14     -!.  .  .  .  .  .  .  .  .  .  .  .!-
```

@MOVEFIL UTILITY DOCUMENTATION

GENERAL INFORMATION

Purpose:

Introduction of the 2200LVP and 2200SVP products presents certain problems concerning file moves, typically not present on other 2200 systems.

1. The diskette and the fixed disk platters differ in size (1MB vs. 4MB).

   Thus, the BASIC-2 MOVE statement cannot be used to move files larger than 1MB from fixed disk to diskette. Furthermore, ISS does not support multi-volume file move. Transport of large files through diskettes thus becomes difficult for the user.

2. The LVP and SVP diskette drive does not support the standard Wang SSSD format diskettes. Transport of software between 2200 VP/MVP and 2200 LVP/SVP is, thus, a problem.

3. Since most vendors, initially at least, will not have access to an LVP or SVP, distribution of vendor software to LVP or SVP customers is a problem.

The @MOVEFIL utility provides a general file move capability, and in particular, addresses each of the problems mentioned above. Specified files are moved from one platter to another; if necessary, a file can span from one output platter to another.

If the specified file doesn't exist on the output platter, a new file will be created, the user will be prompted for a change in file size (default = no change).

The file is then moved and verified and the procedure is repeated for the next file.

In summary, the utility can:

- Create a new file.
- Change the new file size, for new output files.
- Rename a file.
- Overwrite an existing file.

Multi-volume Output Files

If the file to be moved will not fit on the output platter, a file called @SPAN001 is created which occupies the remainder of the output platter. @SPAN001 contains the actual output file name, the actual size and as much of the file as will fit.

The user is then prompted to mount a platter. A file @SPAN002 is created and the file copy continues. If necessary, a third platter with @SPAN003 can be used, and so forth.

Multi-volume Input Files

If the file to be moved is not in the platter index, the utility looks for the file @SPAN001 and determines if the desired input file spans from this platter to another. If so, it is moved with the appropriate prompting for the next platter. The digits in the @SPAN name and the actual file name in @SPANxxx are verified after each new platter is mounted.

Multi-volume Files Format

- A multi-volume file is always a data file. The first sector contains the following information.
- 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: Size of original file.
- Bytes 5-6: Number of sectors used in original file.
- Bytes 7-8: Portion of file contained in current platter (i.e., binary equivalent of last three digits in "@SPANxxx").
- Bytes 9-17: Name of original file.
- All of the above information is recorded using absolute sector addressing (i.e., DATA LOAD BA ....).
- The rest of the file contains a portion of the original file and a standard 2200 file trailer.

Media

1. 2200 Format Platter

   Files are recorded utilizing the standard Wang 2200 disk catalog structure.

   The output platter must have been formatted and scratched before using this utility.

2. 3741 Platter

   For software interchange between 2200VP/MVP systems with 2270A diskette drives and 2200LVP/SVP, 3741 platters can be utilized. IBM 3741 File Format is, however, not utilized. Data is recorded in a format adopted by this utility, hereafter referred to as "@MOVEFIL 3741 format".

"@MOVEFIL" 3741 Format Description:

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 catalog structure.

Restrictions

1. This utility is written in BASIC-2 and will not operate on 2200A, B, C, S, to T systems.

2. The utility does not format or scratch any platters.

3. The move operation terminates if a bad sector is encountered on either the input or output platter.

4. The program requires a 14K partition for execution.

Procedure

Files are moved and verified one at a time. For each file, the user is prompted to enter:

- Input platter type (Wang 2200 or 3741).

- Input platter address.

- Input file name (default = next sequential name).

- Output platter type (Wang 2200 or 3741).

- Output platter address.

- Output file name.

Operating Instructions

To run the program type LOAD RUN "@MOVEFIL" RETURN.

For each file, the user is prompted to enter:

- Input platter type: "W" if the platter is any 2200 format platter; "I" if it is a 3741 platter.

- Input platter address.

- Input file name (default = next sequential name). In the case when the specified name is of the form "@SPANxxx", the last three characters must be "001" (i.e. any other characters in the last three positions will be considered invalid and rejected as such.

- Number of free sectors: The operator is informed of the number of sectors that are currently used in the input file and on how many are free; negative numbers are not allowed. (Default = existent number of free sectors).

- Output platter type: "W" if the platter is any 2200 format platter; "I" if it is a 3741 platter (default = "W").

- Output platter address (default initially is D10, thereafter it is the last entered output platter address).

- Output file name (default = input file name). Names of the form "@SPANxxx" are rejected as invalid. However, such names will be used by the program if necessary (i.e. if the file will not fit entirely in one output platter).

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

0591E