VP BASIC-2 RELEASE 1.9
(Platter 701-2118H)

Release 1.9 of 2200VP BASIC-2 replaces all previous VP BASIC-2 releases. Release 1.9 is required for any VP with 2280 disk drives. This release also provides all VP systems with several new features and corrects all known system anomalies. Listed below are the system changes since Release 1.8. The following files on the system platter differ from Release 1.8:

"@@"	-	VP OS & BASIC-2 interpreter.
"@MOVE"	-	Move system file utility.
"@FORMAT"	-	Format disk utility.
"@A","@B","@C"	-	System diagnostics

A. System Enhancements

The following enhancements are also available with MVP BASIC-2 Release 1.7.

1. 2280 Disk Drives

Release 1.9 supports the 2280 disk drives; previous releases should not be used. For details concerning 2280 operations, see the "2280 Disk Drive User Manual".

2. Disk Platter Formatting

A format disk platter utility, "@FORMAT", for formatting 2260C, 2260BC, and 2280 disk platters resides on the system diskette. To format a disk platter, LOAD RUN "@FORMAT" and respond to the prompts as requested. The format utility makes use of the new $FORMAT DISK statement to format the specified disk platter (see following description of $FORMAT DISK).

3. Variable Device Address Specification

Device addresses used in SELECT statements can now be specified by the value of an alpha variable as well as explicitly by 3 hexdigits. Device addresses in SELECT statements have the following format:

$$\text{device address} = \left\{ \begin{array}{l} \text{hexdigit hexdigit hexdigit} \\ \\ \text{<alpha-variable>} \end{array} \right\}$$

where the value of the alpha variable must be 3 ASCII hexdigits representing the device type and address.

Examples:

A$ = "320": SELECT #3 <A$>   selects #3 to disk 320

A$ = "215": SELECT PRINT <A$> (132)   selects print to device 215 with a line width of 132 characters.

4. System Diagnostics

   The system diagnostics, accessible immediately after power on, have been rewritten to support the larger memory configurations and to provide better memory diagnosing with more consistent error messages and improved displays.

5. MOVE/COPY

   The MOVE and COPY disk operations now make more efficient use of memory for buffering; these operations are now up to 15% faster.

6. $RELEASE PART

   The syntax for the MVP $RELEASE PART statement is supported; however, the statement is ignored if executed.

7. Reloading System Software

   Executing $INIT "SYSTEM" in immediate mode, allows the system to be reloaded with BASIC-2 or a system diagnostic without powering the system off and then back on. When $INIT "SYSTEM" is executed, control is passed to the system bootstrap. The bootstrap message

   ```
   MOUNT SYSTEM PLATTER
   PRESS RESET
   ```

   is displayed and the system can be loaded, as if the system had just been turned on. Note that the program and data in memory are cleared when $INIT "SYSTEM" is executed.

B. Corrected Anomalies

1. If the ERROR statement was used to recover from an error within a subroutine called from a special function key, the system lost the subroutine return information. This would result in ERR P41 when RETURN was executed.

2. The system did not always detect the illegal occurrence of alpha array elements where numeric variables were expected.

3. If variables were used to specify array dimensions (e.g., DIM X(R,C)) during program overlaying, erroneous memory overflow errors (ERR A02) might be reported by the system. This could only happen if the program text being overlayed was larger than that of the overlay loaded, more variables were defined in the overlay than existed in the original program, and most of memory was used.

4. A PACK statement with an exponential image could modify the values of the variables containing the data to be packed. If an exponential image in a PACK statement did not have exactly 1 integer digit specified and the data to be packed was specified by a numeric array designator (e.g., N()), the result of the pack would be correct but the exponents of the values in the numeric array would be changed.

5. MVP global variables (e.g., @X$) were listed by LIST V as local variables (e.g., X).

6. SELECT P timing was approximately 10% fast.

7. $CLOSE disk did not release (unhog) the specified disk unit. $CLOSE /xyy should have been equivalent to $GIO /xyy (4400) disk unhog.

WANG LABORATORIES, INC. G. Castro

Computers 1126

MARKETING RELEASE

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 12
FROM Al Breveleri	DATE May 1979
SUBJECT 2200VP Operating System Release 2.0	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION □ YES □ NO

Wang announces version 2.0 of the 2200VP operating system. This version supports the new model 2280 disk drives (26.8, 53.6, and 80.4 MB), adds new capabilities, increases system performance, and corrects all known firmware errors. Release 2.0 replaces all previous versions and is required for any VP configured with a 2280 disk drive. Note that release 1.9 had a limited distribution, and most users of version 2.0 will be replacing version 1.8.

Features

- 2280 Disk drive support
- $FORMAT DISK statement
- Variable device address specification
- MVP program compatibility

Restrictions

- All VP systems should be upgraded to OS version 2.0.

Ordering Information

Order through the Software Distribution Center, Package number 195-0033-3/8. Distribution media for diskette is platter number 701-2118J, and for mini-diskette, platters numbered 701-8127 for the operating system and 701-8128 for the corresponding diagnostics.

Availability

In stock; allow one week for delivery.

Price

No charge to 2200VP or 2200MVP owners.

(WANG) LABORATORIES, INC.
Computers

MARKETING RELEASE

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 12
FROM Al Breveleri	DATE May 1979
SUBJECT 2200VP Operating System Release 2.0	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION □ YES □ NO

Support

This is a category 1, Wang supported product. Any suspected errors or anomalies found in this package should be reported to Wang Laboratories via the local district analyst.

The following files on the system platter differ from Release 1.8:

"@@"	-	VP OS & BASIC-2 interpreter.
"@MOVE"	-	Move system file utility.
"@FORMAT"	-	Format disk utility.
"@A","@B","@C"	-	System diagnostics

A. System Enhancements

The following enhancements are also available with MVP BASIC-2 Release 1.7.

1. 2280 Disk Drives

Release 2.0 supports the 2280 disk drives; previous releases should not be used. For details concerning 2280 operations, see the "2280 Disk Drive User Manual".

2. Disk Platter Formatting

A format disk platter utility, "@FORMAT", for formatting 2260C, 2260BC, and 2280 disk platters resides on the system diskette. To format a disk platter, LOAD RUN "@FORMAT" and respond to the prompts as requested. The format utility makes use of the new $FORMAT DISK statement to format the specified disk platter (see following description of $FORMAT DISK).

3. Variable Device Address Specification

Device addresses used in SELECT statements can now be specified by the value of an alpha variable as well as explicitly by 3 hexdigits. Device addresses in SELECT statements have the following format:

```
                   hexdigit hexdigit hexdigit
device address =
                      alpha-variable
```

where the value of the alpha variable must be 3 ASCII hexdigits representing the device type and address.

Examples:

A$ = "320": SELECT #3 A$	selects #3 to disk 320
A$ = "215": SELECT PRINT A$ (132)	selects print to device 215 with a line width of 132 characters.

4. System Diagnostics

The system diagnostics, accessible immediately after power on, have been rewritten to support the larger memory configurations and to provide better memory diagnosing with more consistent error messages and improved displays.

5. MOVE/COPY

The MOVE and COPY disk operations now make more efficient use of memory for buffering; these operations are now up to 15% faster.

6. $RELEASE PART

The syntax for the MVP $RELEASE PART statement is supported; however, the statement is ignored if executed.

(WANG) LABORATORIES, INC.

Computers

MARKETING RELEASE

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 12
FROM Al Breveleri	DATE May 1979
SUBJECT 2200VP Operating System Release 2.0	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION □ YES □ NO

7. Reloading System Software

Executing $INIT "SYSTEM" in immediate mode, allows the system to be reloaded with BASIC-2 or a system diagnostic without powering the system off and then back on. When $INIT "SYSTEM" is executed, control is passed to the system bootstrap. The bootstrap message

```
MOUNT SYSTEM PLATTER
PRESS RESET
```

is displayed and the system can be loaded, as if the system had just been turned on. Note that the program and data in memory are cleared when $INIT "SYSTEM" is executed.

B. Corrected Anomalies

1. If the ERROR statement was used to recover from an error within a subroutine called from a special function key, the system lost the subroutine return information. This would result in ERR P41 when RETURN was executed.

2. The system did not always detect the illegal occurrence of alpha array elements where numeric variables were expected.

3. If variables were used to specify array dimensions (e.g., DIM X(R,C)) during program overlaying, erroneous memory overflow errors (ERR A02) might be reported by the system. This could only happen if the program text being overlayed was larger than that of the overlay loaded, more variables were defined in the overlay than existed in the original program, and most of memory was used.

4. A PACK statement with an exponential image could modify the values of the variables containing the data to be packed. If an exponential image in a PACK

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 12
FROM Al Breveleri	DATE May 1979
SUBJECT 2200VP Operating System Release 2.0	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION □ YES □ NO

statement did not have exactly 1 integer digit specified and the data to be packed was specified by a numeric array designator (e.g., N()), the result of the pack would be correct but the exponents of the values in the numeric array would be changed.

5. MVP global variables (e.g., @X$) were listed by LIST V as local variables (e.g., X).

6. SELECT P timing was approximately 10% fast.

7. $CLOSE disk did not release (unhog) the specified disk unit. $CLOSE /xyy should have been equivalent to $GIO /xyy (4400) disk unhog.

$FORMAT DISK

```
! General Form:                                        !
!                           file #                     !
!     $FORMAT DISK platter                             !
!                           disk-address               !
!                                                      !
```

Purpose:

The $FORMAT DISK statement issues a command to the disk processing unit to format the specified disk platter. This statement can only be used with disks that support formatting under software control (e.g., 2260C, 2260BC, 2280). Formatting on certain 2200 disks (e.g., 2270) is initiated by pressing the format button located on the disk unit.

Before a platter can be used for the storage and retrieval of data by the user, the platter must be formatted. Formatting involves recording a unique address for each sector on the disk platter, along with other control information used by the disk processing unit when accessing a sector. All data within the sector is zeroed.

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 12
FROM Al Breveleri	DATE May 1979
SUBJECT 2200VP Operating System Release 2.0	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION □ YES □ NO

```
+--------------------------------------------------------+
! Caution:  Formatting a disk platter overwrites all    !
! data that may previously have been stored on the      !
! platter.  It is recommended that the Wang supplied    !
! format utility be used, to prevent the accidental     !
! formatting of the wrong disk platter.                 !
+--------------------------------------------------------+
```

Examples of valid syntax:

```
10 $FORMAT DISK T/310
20 $FORMAT DISK T/D11
30 $FORMAT DISK R#2
```

VP BASIC-2 RELEASE 2.0
(Diskette Platter 701-2118J)
(Minifloppy Platters 701-8127 BASIC-2 and 701-8128 Diagnostics)

Release 2.0 of BASIC-2 replaces all previous VP BASIC-2 releases. This release fixes one problem introduced in Release 1.9. Since Release 1.9 was not widely distributed, the documentation for Release 1.9 is included here. All description of Release 1.9 applies also to Release 2.0.

The following files on the system platter(s) differ from Release 1.9:

"@@" - VP OS & BASIC-2 interpreter
"@MOVE" - Move system file utility.

A. Corrected Anomaly

1. LIST T did not work in Release 1.9. This has been corrected in Release 2.0.

[illegible] / 11[illegible]

(WANG) LABORATORIES, INC.

Computers

MARKETING RELEASE

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 14
FROM Tom Camp	DATE July 1979
SUBJECT VP Operating System Release 2.1	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION □ YES □ NO

Wang proudly announces version 2.1 of the 2200VP Operating System. This new release provides features to utilize Wang's new exciting interactive terminal, the 2236DE.

Features

- New function PRINT BOX for 2236DE.

Restrictions

- All 2200VP systems should be upgraded to Release 2.1.

Ordering Information

Order through Software Distribution:

Package Number: 195-0033-3
Diskette Number: 701-2118K

Availability

In stock, allow one week for delivery.

Price

No charge to 2200VP owners.

Support

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

VP BASIC-2 RELEASE 2.1
(Diskette Platter 701-2118K)
(Mini-diskette Platter 701-8127A)

Release 2.1 of 2200VP BASIC-2 replaces all previous VP BASIC-2 releases. This release provides all VP systems with several new features and corrects all known system anomalies. Listed below are the system changes since Release 2.0. The following file on the system platter differs from Release 2.0:

"@@" - VP OS & BASIC-2 interpreter

A. SYSTEM ENHANCEMENTS

1. PRINT BOX Function

A new function, BOX (X, Y), has been added to the PRINT statement for drawing or erasing lines and boxes on terminals having box graphics capability (such as the 2236DE). The BOX function has also been added to 2200MVP BASIC-2 Release 1.8.

B. CORRECTED ANOMALIES

1. Immediate Mode load no longer allows the BEG parameter in its syntax since it has no function.

2. ON ERROR GOTO can now be used to trap D82 (file not found) errors occuring because of an attempt to overlay in a non-existent file.

3. Accessing a disk using a device type other than 3, B, or D now results in ERR P48.

WANG LABORATORIES, INC.

Computers

MARKETING RELEASE

TO DISTRIBUTION	PUBLICATION #
FROM JOHN SULLO	DATE FEBRUARY 1980
SUBJECT VP OPERATING SYSTEM RELEASE 2.2	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION □ YES □ NO

Wang Laboratories, Inc. announces version 2.2 of the 2200VP Operating System. This release provides several new and features.

FEATURES

- Programmable LOAD RUN command.
- 2273 Band Printer vertical format control.

RESTRICTIONS

- All 2200VP systems should be upgraded to Release 2.2.

ORDERING INFORMATION

Order through Software Distribution:

Package Number:	195-0033-3
Diskette Number:	701-2118L

Mini-diskette version:

Package Number:	195-0033-8
Diskette Numbers:	701-8068D
	701-8069C (Diagnostics)

AVAILABILITY

In stock, allow one week for delivery.

PRICE

No charge to 2200VP owners.

SUPPORT

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

NOTE: Attached for your convenience is a section describing the enhancements and corrections released in prior versions of the Operating System.

VP BASIC-2 RELEASE 2.2
(Diskette Platter 701-2118L)
(Mini-diskette Platters 701-8068D, 8069C (Diagnostics))

Release 2.2 of 2200VP BASIC-2 replaces all previous VP BASIC-2 releases. This release provides all VP systems with several new features and corrects all known system anomalies. The system platter includes the VP Operating System and BASIC-2 language processor, system diagnostics, and several utilities.

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 2200VP configurations.

@MOVE: Move System Files

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

@MENU: Program Menu

@MENU provides a tree structured menu 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, and 2280 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)

II. SYSTEM CHANGES

Listed below are the system changes since Release 2.2. The following files on the system platter differ from Release 2.2:

@@ -	VP OS & BASIC-2 interpreter
@MENU -	Menu Utility (new)
.SYSVPB -	Menu Mode Data (new)
.STARTD -	Menu Root Mode Pointer (new)
@2273VFU -	2273 Vertical Format Control (new)

A. SYSTEM ENHANCEMENTS

LOAD RUN

The LOAD RUN command is now programmable and can be used to clear out an executing program and load in another program. It has the advantage over the LOAD statement in that program dependent parameters, including program protect, are cleared.

B. CORRECT ANOMALIES

1. Certain types of 2280 disk errors (e.g., illegal sector address or disk in load mode) could cause the wrong sectors to be reported in error during VERIFY operations.
2. VERIFY operations did not set the ERR function if an error was detected.
3. Printing a numeric value in exponential format with PRINTUSING could produce incorrect output if the number of digits specified in the image was greater than 13.
4. Halt/stepping a multi-statement line containing a LIST V statement could cause previously executed statements to be re-executed.
5. Spaces in the value of an alpha variable used for device selection in a SELECT statement were ignored, but should have been reported as an ERR 17.
6. EXP function caused the ROUND/NOROUND flag to be reset to the ROUND state. The code now will restore the flag to the selected state.
7. Using the same array as both a receiver and an operand in a MAT multiply or transpose statement was not reported as an error. Furthermore, using a global array and a local array which had the same name letter (and digit) was incorrectly reported as an error.
8. If an error occurred while opening a data file with a device table slot (#n) that already had a file open, the previous file would not be closed.

The attached pages contain system enhancements and corrections for previous releases of the Operating System.

Changes are shown separately and in detail for each of the last four releases; another section describes changes prior to Release 2.2.

RELEASE 2.1

System Enhancements:

1. PRINT BOX Function
   A new function, BOX (X,Y) has been added to the PRINT statement for drawing or erasing lines and boxes on terminals having box graphics capability (such as the 2236DE - see 2236DE Reference Manual).

Corrected Anomalies:

1. Immediate Mode Load no longer allows the BEG parameter in its syntax since it has no function.
2. ON ERROR GOTO can now be used to trap D82 (file not found) errors occurring because of an attempt to overlay in a non-existent file.
3. Accessing a disk using a device type other than 3, B, or D now results in ERR P48.

RELEASE 2.0

Corrected Anomaly:

LIST T did not work in Release 1.9. This has been corrected in Release 2.0.

RELEASE 1.9

System Enhancements:

The following enhancements are also available with MVP BASIC-2 Release 1.7.

1. 2280 Disk Drives

   Release 1.9 supports the 2280 disk drives; previous releases should not be used. For details concerning 2280 operations, see the "2280 Disk Drive User Manual".

2. Disk Platter Formatting

   A format disk platter utility, "@FORMAT", for formatting 2260C, 2260BC, and 2280 disk platters resides on the system diskette. To format a disk platter, LOAD RUN "@FORMAT" and respond to the prompts as requested. The format utility makes use of the new $FORMAT DISK statement to format the specified disk platter.

$FORMAT DISK

```
! General Form:                                              !
!                                                            !
!   $FORMAT DISK platter       file#                         !
!                              disk-address                  !
```

Purpose:

The $FORMAT DISK statement issues a command to the disk processing unit to format the specified disk platter. This statement can only be used with disks that support formatting under software control (e.g., 2260C, 2260BC, 2280). Formatting on certain disks (e.g., 2270) is initiated by pressing the format button located on the disk unit.

Before a platter can be used for the storage and retrieval of data by the user, the platter must be formatted. Formatting involved recording a unique address for each sector on the disk platter, along with other control information used by the disk processing unit when accessing a sector. All data within the sector is zeroed.

```
!Caution: Formatting a disk platter overwrites all !
!data that may previously have been stored on the  !
!platter.  It is recommended that the Wang supplied!
!format utility be used, to prevent the accidental !
!formatting of the wrong disk platter.
```

Examples of valid syntax:

```
10 $FORMAT DISK T/310
20 $FORMAT DISK T/D11
30 $FORMAT DISK R#2
```

3. Variable Device Address Specification

Device addresses used in SELECT statements can now be specified by the value of an alpha variable as well as explicitly by three hexdigits.

Device addresses in SELECT statements have the following format:

$$\text{device} = \left\{ \begin{array}{c} \text{hexdigit hexdigit hexdigit} \\ \\ \text{alpha-variable} \end{array} \right\}$$

where the value of the alpha variable must be three ASCII hexdigits representing the device type and address.

Examples:

$A = "320": SELECT #3 <A$>	selects #3 to disk 320
A$ - "215": SELECT PRINT <A$>(132)	selects print to device 215 with a line width of 132 characters

4. System Diagnostics

The system diagnostics, accessible immediately after power on, have been rewritten to support the larger memory configurations and to provide better memory diagnosing with more consistent error messages and improved displays.

5. MOVE/COPY

The MOVE and COPY disk operations now make more efficient use of memory for buffering; these operations are now up to 15 percent faster.

6. $RELEASE PART

The syntax for the MVP $RELEASE PART statement is supported; however, the statement is ignored if executed.

7. Reloading System Software

Executing $INIT "SYSTEM" in immediate mode, allows the system to be reloaded with BASIC-2 or a system diagnostic without powering the system off and then back on. When $INIT "SYSTEM" is executed, control is passed to the system bootstrap. The bootstrap message

```
MOUNT SYSTEM PLATTER
PRESS RESET
```

is displayed and the system can be loaded, as if the system had just been turned on. Note that the program and data in memory are cleared when $INIT "SYSTEM" is executed.

Corrected Anomalies

1. If the ERROR statement were used to recover from an error within a subroutine called from a special function key, the system lost the subroutine return information. This resulted in ERR P41 when RETURN was executed.
2. The system did not always detect the illegal occurrence of alpha array elements where numeric variables were expected.
3. If variables were used to specify array dimensions (e.g., DIMX(R,C)) during program overlaying, erroneous memory overflow errors (ERR A02) might be reported by the system. This could happen only if the program text being overlayed were larger than that of the overlay loaded, more variables were defined in the overlay than existed in the original program, and most of memory was used.
4. A PACK statement with an exponential image could modify the values of the variables containing the data to be packed. If an exponential image in a PACK statement did not have exactly one integer digit specified and the data to be packed were specified by a numeric array designator (e.g., N()), the result of the pack would be correct but the exponents of the values in the numeric array would be changed.

5. MVP global variables (e.g., @X$) were listed by LIST V as local variables (e.g., X).
6. SELECT P timing was approximately 10 percent fast.
7. $CLOSE disk did not release (unhog) the specified disk unit. $CLOSE /xyy should have been equivalent to $GIO /xyy (4400) disk unhog.

RELEASE 1.8

System Enhancements:

1. MVP Statements

   Release 1.8 supports all 2200MVP BASIC-2 language features. Thus, MVP programs can be loaded and executed on the 2200VP. The MVP statements behave as if on a single partition, single terminal MVP with the following exceptions:

   a. ERR A08 results if $INIT is executed.
   b. $OPEN and $CLOSE can be used to hog or unhog a disk if the disk address is specified; the device type specified must be 3 or B. $CLOSE with no parameters will NOT release all hogged disks, as it does on the MVP.
   c. There is no broadcast message. The $MSG statement can be executed but the message will not be displayed when RESET is pressed. The $MSG function returns a value of HEX (00).

2. LOAD RUN

   The default platter specification for LOAD RUN has been changed from 'F' to 'T'; that is, the platter to load from is specified by the device type in the #0 slot of the device table. For example,

   ```
   :SELECT DISK B10
   :LOAD RUN
   ```

   loads and executes the program "START" from the removable platter of the 10MB drive.

3. New $GIO Microcommands

   The $GIO command repertoire as been expanded to include:

C3h_3h_4 -- same as C6h_3h_4 except WR before CPB
81h_3h_4 -- same as 80h_3h_4 except WR before CPB
83h_3h_4 -- same as 82h_3h_4 except WR before CPB
870 0 -- same as 860 0 except WR before CPB
870 h_4 -- same as 860 h_4 except WR before CPB
89h_3h_4 -- same as 88h_3h_4 except WR before CPB
8Bh_3h_4 -- same as 8Ah_3h_4 except WR before CPB

4. ERR P48

Error #48 (Illegal Device Address) is now recoverable using the ERROR statement.

5. CONTINUE

CONTINUE is now legal in a multi-statement immediate mode line; however, it must be the last statement on the line.

Corrected Anomalies

1. Attempting to save a program or data file on a platter with 0 index sector (e.g., a platter which has just been formatted) now results in ERR D85.
2. The editor no longer turns the cursor off and on during cursor manipulation in Console Input, INPUT and LINPUT operations. This is important for certain remote terminals connected to a non-buffered communications controller.

The following language enhancements were made in earlier releases:

1. The ELSE clause may be used after an ON statement.

2. The MAT SEARCH statement permits the use of a literal in place of the search-alpha-variable.

3. The POS function allows the use of a literal in place of the alpha-variable to be searched.

4. LINPUT has been expanded to make the automatic entry into Entry Mode optional.

5. LIST T provides a cross-reference listing of program lines that contain a specified character string.

```
1.        (alpha) (SELECT)
     ON   { or  } {GOSUB }      argument-list  :  ELSE statement
          (expr ) (GOTO  )
```

The ON statements have been expanded to allow ELSE clauses. The expression is evaluated, then the argument list is scanned for the specified element. If the expression is zero, too large, or points to a null argument, the ON statement effectively becomes equivalent to an IF statement with a false condition. In this case, the ELSE clause is executed.

Example:

```
10 DIM A$1
•
•
•
50 ON A$ SELECT R;D;;G
   :ELSE GOTO 100
```

This will select R, D, or G if A$ = HEX(01), 02, or 04, respectively. For all other values, the ELSE clause will cause a branch to line 100.

Examples of valid syntax:

```
ON C(I) GOSUB 100, 500,,,100
:ELSE GOSUB 1000

ON STR(B$,7,1) SELECT DISK 310; DISKB10
:ELSE LOAD F/320, "START"
```

```
2.             ⎧search-alpha variable⎫   ⎧ <  ⎫ ⎧alpha-variable⎫
    MAT SEARCH ⎨  or                 ⎬,  ⎪ <= ⎪ ⎨  or          ⎬  TO
               ⎩search-literal       ⎭   ⎨ =  ⎬ ⎩literal       ⎭
                                         ⎪ >  ⎪
                                         ⎪ >= ⎪
                                         ⎩ <> ⎭

      locator-alpha variable [STEP S]
```

The MAT SEARCH has been expanded to allow the use of literals in place of the search alpha variable. This form is useful when a fixed table needs to be searched for variable data. It saves assigning the data to another array, and it results in code that is more self-explanatory.

Example:

```
10 DIM A$3, C$2
50 LINPUT "DEVICE ADDRESS", A$
   :MAT SEARCH "310B10320B20350B50360B60", = STR(A$,,3) TO C$ STEP 3
   :ON 1 + VAL(C$,2)/3 SELECT #N/310;#N/B10; #N/320; #N/B20; #N/350;
   #N/B50; #N/360; #N/B60
   :ELSE GOTO 50
   .
   .
   .
```

This example takes in a 3 byte device address from the operator, verifies it against a list of valid addresses, and selects the request device. If the address was invalid, it branches back.

Examples of valid syntax:

```
MAT SEARCH "LDASTAADC", = STR(Z$,1,3) TO B$() STEP N

MAT SEARCH A$() <1, 400 > , = "ABC" TO B$

MAT SEARCH HEX(0020413749), = B$ TO C$
```

```
                              <
                              <=
3.          alpha-variable    =     alpha variable
   POS ([-] literal           >     literal string       )
                              >=    hh
                              <>
```

The POS function has been expanded to allow the use of literals in place of the first alpha variable. This form is useful when a fixed table of characters needs to be compared to a single character (in a variable). It saves assigning the data to another variable, and it results in code that is more self-explanatory.

Example:

```
10 DIM A$1
20 LINPUT "LOAD, SAVE, VERIFY, OR DISPLAY", A$
30 ON POS ("LSVD" = A$) GOSUB 100, 200, 300, 400
   :ELSE GOSUB 500
    .
    .
    .
```

This example easily separates the valid letters (L, S, V, and D) from the invalid, and calls routines to handle the five cases.

Examples of valid syntax:

```
T = POS("123456789" = T$)

J = POS (HEX(0102040810204080)    A$)

L = POS(-"0:A a "    = STR(B$,N,L))
```

4. LINPUT [literal] [,] [?] [-] alpha-variable

The LINPUT statement has been expanded to make the automatic entry into Edit Mode optional. If a '?' is included in the LINPUT statement, the LINPUT operation will begin in Text Entry Mode rather than Edit Mode and the cursor will be positioned after the last non-blank character in the field rather than at the beginning of the field. LINPUT? is useful when special function keys are to be used to control field operations (i.e., the Edit key need not be pressed prior to pressing the special function key in order to enter the user subroutine).

Example:

```
:10 DIM A$5
:20 A$="ABC"
:30 LINPUT "EDIT VALUE" -A$
:RUN
ENTER VALUE * A B C _ _
```

```
:10 DIM A$5
:20 A$="ABC"
:30 LINPUT "EDIT VALUE"?-A$
:RUN
ENTER VALUE A B C _ _
```

LIST T

General Form

$$\text{LIST [S] [title] T}\left\{\begin{array}{l}\text{literal}\\ \text{alpha-variable}\end{array}\right\}\left[,\ \left\{\begin{array}{l}\text{literal}\\ \text{alpha-variable}\end{array}\right\}\right]\ \ldots$$

where:

$$\text{title} = \left\{\begin{array}{l}\text{literal}\\ \\ \text{alpha-variable}\end{array}\right\}$$

Purpose:

The LIST T command generates a cross-reference listing of all program text lines that contain the specified string of characters. The string to be searched for can be specified by a literal or the value of an alpha variable. Space characters are ignored in both the string being searched for and program text. More than one string can be searched for by including more than one literal or alpha-variable in the LIST T argument list. The LIST T command is programmable.

The 'S' parameter is a special feature for the CRT display. It permits the listing of the cross-reference in steps; that is, listing stops when the screen is full. To continue listing, (EXEC) must be keyed. Note that for nonstandard CRT's, the number of lines on the CRT is specified by a SELECT LINE statement. The 'S' parameter is ignored in Program Mode.

Keying HALT/STEP during the listing of a cross-reference stops the listing after the current line is listed (however, the listing cannot be continued). Alternatively, the operator may slow down listing on the CRT by invoking a pause of from 1/6 to 1 1/2 seconds with a SELECT P statement. In this case, the system pauses for the specified interval after each line is displayed.

The optional "title" parameter is a convenient means of identifying hardcopy cross-reference listing. If a title is included in the LIST T command, the system performs the following actions:

a. Issues a top-of-form (HEX(OC)) code.
b. Prints the title in expanded print.
c. Skips a line.
d. Prints the listing.

On a printer, these actions cause the title to be printed at the top of page in expanded print, followed by a blank line and the cross-reference listing. On a CRT, the title is displayed before the cross-reference listing.

When the system is Master Initialized, the CRT is initially selected for LIST Operations. Other printing devices may be selected for listing with a SELECT LIST statement (See SELECT).

EXAMPLES:

```
:LISTD
0010 REM DISK SELECT IS ON LINE 20
0020 SELECT DISK /320, #3 /B20
0030 DATA LOAD DC OPEN T#3,"PEANUT"
   : DSKIP #3, 1
0040 LOAD F#3,"BUTTER"

0050 REM LIST T IGNORES S PA  C  E S

:LIST T "SELECT"
"SELECT"
     -   0010 0020

:LIST T "#3"
"#3"
     -   0020 0030 0040

:A$="SP  A C   E S":LIST T A$
:SP  A C  E S"
     -   0500

:LIST T "A", "B", "C", "Z"
"A"
     -   0300  0040  0050
"B"
     -   0020  0040
"C"
     -   0010  0020  0030  0050
"Z"
```

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     -!.  .  .  .  .  .  .  .  .  .  .  .!-
```

After the input has been specified (# lines, bottom of form), you can edit the format by setting the desired tab stops (or deleting unwanted ones). Note: Tab stops are not allowed in Channel 1 or Line 1 - these are reserved for top and bottom of form. Be sure to set the printer to correspond to the number of lines per inch specified in the format definition.

Once defined, the format can be tested by specifying the number of the channel to be tested. The program skips to, and prints a line of text for, each tab stop on the specified channel. When the test is satisfactory, the format may be saved on a data file. When the file already exists, it will be overridden.

WANG LABORATORIES, INC

Computers

MARKETING RELEASE

TO DISTRIBUTION	PUBLICATION #
FROM MARKET PLANNING AND DEVELOPMENT	DATE AUGUST 1980
SUBJECT VP/SVP OPERATING SYSTEM RELEASE 2.3	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION □ YES □ NO

Wang Laboratories announces version 2.3 of the single user VP Operating System. This release provides several new features:

FEATURES

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

RESTRICTIONS

- All 2200VP systems should be upgraded to Release 2.3.
- All 2200VP systems require Release 2.3.

ORDERING INFORMATION

Order through Software Distribution:

	2200 VP	2200 SVP	Minidiskette
Package Number	195-0033-3	195-2163-5	195-0033-8
Diskette Number	701-2118M	704-0001A	701-8068E (OS/Languages) 701-8069D (Diagnostic) 701-8129 (Utilities)

AVAILABILITY

In stock, allow one week for delivery.

PRICE

No charge for 2200VP, 2200SVP 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.

VP BASIC-2 RELEASE 2.3

This release is available on the following media:

1. Part # 701-2118M: SSSD Diskette (2270, 2270A).
2. Part # 704-0001A: DSDD Diskette (2200SVP).
3. Part # 701-8068E, 701-8069D and 701-8129: Mini Diskettes (2210).

Release 2.3 of VP BASIC-2 replaces all previous VP BASIC-2 releases. This release provides all 2200VP and SVP systems with several new features and corrects all known system anomalies. The system platter includes the VP (single-user) Operating System and BASIC-2 language processor, system diagnostics, and several utilities.

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.

@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 2.2. The following files on the system platter differ from Release 2.2:

@@	-	VP OS & BASIC-2 interpreter
@MENU	-	Menu Utility
@BACKUP	-	Platter Backup (new)
@RECOVER	-	Backup Recovery (new)
@MOVEFIL	-	Move File (new)
@MOVE1	-	Move File overlay (new)
.SYSVPB	-	Menu Mode Data
@MOVE	-	Move System Files
@2273VFU	-	2273 Vertical Format Control
@FORMAT	-	Format Disk Platter

A. SYSTEM ENHANCEMENTS

1. VP BASIC-2 now supports the 2200SVP as well as the 2200VP hardware.

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

B. CORRECT ANOMALIES

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

2. The system did not terminate multi-sector write operations (DATASAVE DC, DATASAVE DA) properly for 2280 disk units. If RESET, 2280 DPU power off, or cartridge dismount were performed, the last sector(s) of the multi-sector write might not be written from the 2280DPU memory to the disk. This was not usually the case since any other disk operation would cause a proper termination of the multi-sector write.

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

4. The 2273 Vertical Format Control utility, @2273VFU, did not treat bottom of form properly. Th 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.

@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 <u>if available</u> 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 rexponse 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$(128)
DATA LOAD DC OPEN T 'filename' : REM 'filename' = name of the user file
DATA LOAD DC T$()
$GIO/xyy' (A000,G$) 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 <u>must</u> 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.

WANG LABORATORIES, INC.

Benincasa, M. 3118

TO: 2200 Series User

From: Wang Laboratories 2200 Development

Subject: New releases of the 2200 MVP and 2200 VP Operating Systems

Date:June 15,1981

Dear 2200 User:

Wang is pleased to supply you with the latest release of the 2200 Series Operating System. Along with the System Diskette should be enclosed a Marketing Release. This describes the changes and features of this release of the Operating System. We are also enclosing an updated System Utilities User Manual that describes the operation and use of all of the utilities that reside on the the System Disk.

If you have questions or problems with this new release of the Operating System, please call your local Wang Analyst.

2460E

WANG LABORATORIES, INC

Computers

MARKETING RELEASE

TO DISTRIBUTION	PUBLICATION #
FROM 2200 PRODUCT DEVELOPMENT	DATE JUNE 1981
SUBJECT VP SINGLE-USER OPERATING SYSTEM RELEASE 2.4	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION □ YES □ NO

Wang Laboratories is pleased to announce release 2.4 of the VP Single-user Operating System. This release includes several new features and enhancements to earlier versions of the VP Single-user Operating System.

SUMMARY OF FEATURES

- *The Operating System Loading Sequence has been modified to allow for easier use of the system diagnostics.*
- *Many of the System Utilites have been improved to run faster, and to be more user oriented. Among these utilities are @BACKUP, @RECOVER, and @MOVEFIL.*
- *A new BASIC-2 Statements have been added to the operating system. This statement is #ID.*

REQUIREMENTS

- *All 2200VP and 2200SVP sytems should be upgraded to Release 2.4 of the VP Single-user Operating Systems.*

ORDERING INFORMATION

Order through Software Distribution.

	2200 VP	2200 SVP	Minidiskette
Package Number	195-0033-3	195-2163-5	195-0033-8
Diskette Number	701-2118N	704-0001B	701-8068F 701-8069E 701-8129A

AVAILABILITY

In Stock, allow one week for delivery.

PRICE

No charge to 2200VP, 2200SVP 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.

VP (Single-User) BASIC-2 RELEASE 2.4

Note: The mini diskette release does not support the '@INSTALL' utility. Release 2.4 of VP BASIC-2 replaces all previous VP BASIC-2 releases. This release provides all 2200 VP and 2200 SVP systems with several new features and corrects all known system anomalies. The system platter includes the VP (single-user) Operating System and BASIC-2 language processor, system diagnostics, and several utilities.

I. System Utilities

Significant changes have been made to several of the system utilities. The 2200 BASIC Utilities Manual (preliminary copy attached) should be consulted for an up-to-date description.

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.

@INSTALL:

This utility moves the system files from the release diskette onto a destination disk of the user's choice.

@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 LVP fixed platters. (Refer to the appropriate disk reference manual for detailed formatting information).

@DAVFU: Vertical Format Control

This utility defines printer vertical format control sequences. (See the appropriate printer reference manuals). This utility was previously named @2273VFU.

@BACKUP: Platter Backup

This program provides a multi volume platter backup capability. It is particularly useful for systems, such as 2200 LVP's or 2200 SVP'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 or all active files from one platter to another. If files 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.

@MRTIAN: System Game

This program allows a user to play one of the latest space war games.

II System Changes for VP

Listed below are the system changes since Release 2.3. The following files on the system platter differ from Release 2.3.

@SYSVPB	-	Menu Mode Data (name changed from .SYSVPB)
@@	-	Microcoded Menu Program (new)
@MRTIAN	-	System Game (new)
@MOVE	-	Move System Files (removed)
@INSTALL	-	'ove System Files (new)
@DAVFU	-	Vertical Format Control (replaces @2273VFU)
@BACKUP	-	Platter Backup (improved)
@BOOT	-	Menu Node Bootstrap (new)
@MOVEFIL	-	Move File
@VP	-	VP OS and BASIC-2 Language Processor (new)
@MENU	-	Menu Utility
@DG	-	System Diagnostics Menu
@HELP	-	System Help Documentation (removed)

Wang Single-User BASIC-2 for 2200 VP or 2200 SVP
Release 2.4

A. Corrected Anomalies

1. The work buffer used during the execution of an INPUT statement was not properly freed when an error X73 or X75 was fielded with :ERROR. Over a long period of time, this could lead to a memory full error, A01.

2. The 2200 T BASIC form of the COPY statement (COPY FR) has been corrected to work properly with the 2200 SVP diskette controller and to use the optimal copying strategy when used with the 2280 disk drive.

3. The sector address of the sector in error returned during the execution of a VERIFY statement was incorrect if a disk hardware error (I90 or I91) was encountered.

4. Under some circumstances, it was possible to type off the right end of a field being entered via a LINPUT statement. This resulted in unpredictable modification to variables other than the variable being LINPUT.

5. Global variables may now be used to dynamically dimension arrays (i.e., DIM A$(@X)). As with all dynamic dimensions in BASIC-2, the global variable used to dimension an array or give the length of an alpha variable must be scalar and must be common.

B. Re-definition of Existing Features

1. The INPUT statement can no longer generate the unrecoverable error, S23, during program execution. The only errors INPUT can now generate during execution are X73 and X75, both of which are recoverable with :ERROR.

2. The $MSG function now returns all spaces. Previously it returned a HEX(00). This makes the default value of $MSG consistent between the VP and the MVP operating systems. $MSG cannot be set to a user defined value under the VP operating system.

3. $BREAK and $BREAK! may now be interrupted by the occurrence of a programmable (SELECT ON) interrupt. When an interrupt occurs, execution proceeds to the interrupt handling subroutine. Upon return from the interrupt handling subroutine, execution continues with the statement following the $BREAK statement. $BREAK is the only statement that is either interrupted or terminated by the occurrence of a user defined interrupt. In all other cases, interrupts may occur only after the completion of the currently executing statement and before beginning the next.

C. New Features

1. A new numeric function, #ID, returns the CPU identification number. Each 2200 CPU is assigned a number (a random integer between 1 and 65535) at the time of manufacture. Machines produced prior to the implementation of this feature return a value of 0, but such machines can be field upgraded to have non-zero # ID's. CPU ID's are not guaranteed to be unique, but it is highly unlikely two given machines will have the same number.

This function allows software to tell one CPU from another. The ability to distinguish one CPU from another is useful in restricting software to specific installations and in telling one CPU from another when disk multiplexers are used.

An application program can inhibit program execution if an unknown or unacceptable identification number is read.

In one or more critical sections of the application software (e.g., menu, key routine) a check can be performed to ascertain that the software is executing on the prescribed machine. The check would be of the type:

```
     IF #ID        machine-id# THEN STOP "!@?$"
```

Of course, the section(s) of code performing this check must be scramble protected (i.e., SAVE!) in order to maintain security. Scramble protect inhibits program examination on disk and after loading.

2. Support has been added for the new $ALERT inter-partition signal construct introduced in Release 2.2 of the MVP operating system. ALERT and SELECT ON ALERT are not functional under the VP operating system, but will not generate errors if included in VP BASIC-2 programs.

3. The editor has been enhanced to accept the cursor positioning keys on the 2236DW terminal during program editing, INPUT, and LINPUT.

WANG LABORATORIES, INC.

MEMORANDUM

TO: John Thibault

FROM: Pete Seymour

DATE: February 18, 1982

cc: D. Angel
R. Droz
B. Patterson
J. Sevigny

SUBJECT: BASIC-2 RELEASE 2.5 ANNOUNCEMENT

GENERAL INFORMATION:

All 2200 VP and 2200 SVP single user systems should be updated to Release 2.5 of the VP Single-user Operating System.

Diskette Numbers are:
701-21180 for 2200 VP
704-0001C for 2200 SVP
701-8068G for Mini-diskettes
701-8069F for Mini-diskettes
701-8129B for Mini diskettes

CHANGES AND CORRECTIONS TO REL 2.4

Operating System

In the editor, a problem relating to pressing the ERASE key at the end of a LINPUT field has been corrected.

In the editor, a problem relating to recalling text lines while in non-edit mode on a 2236DW terminal has been corrected.

A problem with the INPUT statement has been fixed. Attempting to supply a value enclosed by quotes in response to an INPUT statement variable list would occasionally result in an ERR X75.

A problem with I96 errors on Phoenix, Winchester, and DSDD diskettes has been fixed by retrying three times on the occurence of such errors.

Utilities:

@DAVFU: The documentation displayed at the beginning of the utility has been changed to show the correct loading sequence of all VFU files created by the utility.

@MOVEFIL: All keyboard entries have been made case replies will no longer result in an incorrect response from the utility.

@BACKUP: An erroneous message concerning the @BADSCTR file has been corrected.

@FORMAT: The program has been changes to control the occurence of all disk errors, and to correctly erase the 15-second message on the terminal.

Diagnostics

The '@-backslash' file as well as the '@A' diagnositc have been updated. These changes were made to update the field service diagnostic menu and to fix a problem with the CPU diagnostic.

FS/lmd

5459A

BASIC-2 Multi-User Operating System Release 2.3

Release 2.3 of the BASIC-2 multi-user operating system has been issued to provide support for the additional memory banks available on the MVPC and the LVPC systems. The LVPC and the MVPC will support BASIC-3/COBOL/DMS. Release 2.3 can run on the MVP, LVP and SVP as well. Diskettes for this upgrade can be obtained from the Software and Literature Control Center.

Order numbers are:

195-2163-3 for the MVP
195-2162-5 for the LVP

Changes and Corrections to Release 2.2:

The Operating System now provides up to 512K bytes of user memory available for the MVPC and the LVPC systems. The maximum number of terminals is still 13 and the maximum number of partitions is still 16.

The Partition Generator (@GENPART) and Partition Status (@PSTAT) System Utilities now support 512K of memory.

Revised system diagnostics for testing control memory and user memory are provided.

On earlier versions of the operating system, the DSKIP and DBACKSPACE statements, executed with the number of sectors or "BEG" (go to the beginning of the file) specified as opposed to the number of logical records specified, could cause the Model 2230MXA/MXB Disk Multiplexer to enter "hog" mode, thus locking out other CPU's. These statements have been corrected for Release 2.3 to enable the disk controller only when it is necessary to read the disk.

Users who utilize $GIO to "hog" and "unhog" the disk are cautioned that a problem similar to the above may occur in their programs if the "unhog" command (CBS 00) is issued without waiting for READY. All users are encouraged to utilize the $OPEN and $CLOSE statements when exclusive control of any device is required.

In addition to @MVP (the MVP Operating System and BASIC-2 Language Processor included in Release 2.3), the following files on the system platter have been changed since Release 2.2. The purpose and operation of each utility remains the same. Refer to the BASIC-2 Utilities Reference Manual (700-6855) for information on each system utility.

@A	CPU Diagnostic
@B	Control Memory Diagnostic
@C	Data Memory Diagnostic
@	Field Service Diagnostics Menu
@P	User Diagnostics Menu
@GENPART	Partition Generator Utility
@PSTAT	Partition Status Utility
@INSTALL	System Installation Utility
@SYSFILE	System Configuration File

2200 BASIC-2 OPERATING SYSTEM RELEASE ANNOUNCEMENT

BASIC-2 Single-User Operating System Release 2.5:

All 2200 VP/SVP single-user operating systems should be updated to Release 2.5. Diskettes are available from the Software and Literature Control Center, 11 Elizabeth Drive, Chelmsford, MA 01824, telephone 800-343-0440 or 617-256-1200, extension 4999.

Order numbers are:

195-2163-3 for the VP
195-2163-5 for the SVP

Changes and Corrections to Release 2.4:

Operating system:

- A problem related to pressing the ERASE key at the end of a LINPUT field has been corrected.

- A problem related to recalling text lines while in non-edit mode on a 2236DW terminal has been corrected..

- A problem with the INPUT statement has been corrected. Attempting to supply a value enclosed by quotes in response to an INPUT request would result in an error X73. Also, not supplying values for all variables in the INPUT statement variable list would occasionally result in an error X75.

Utilities:

- In the @DAVFU utility, the documentation displayed at the beginning of the utility has been changed to show the correct loading sequence for all VFU (vertical format unit for formatting printed output) files created by the utility.

- In the @MOVEFIL utility, all keyboard entries have been made upper/lower-case insensitive; the input of lower case characters will no longer result in an incorrect response from the utility.

- In @BACKUP an erroneous message concerning the @BADSCTR file has been corrected.

- The @FORMAT program has been changed to control the occurrence of all disk errors and to erase the 15 second message on the terminal correctly.

Diagnostics:

- The '@ ' file and the '@A' diagnostic have been updated to fix a problem with the CPU diagnostic.

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

COMPANY CONFIDENTIAL

Software Release Notice # 0099

Software Engineering

WANG

		ECO No. 28011	*Effective Date* 05/26/83
Description Basic-2 Single User Operating System			*Version* 2.6
Product Line 2200 VP	*Release Type* Maintenance		*Package Number* 195-0033-X

Purpose of Release Changes have been made to the Operating System and to some of the utilities.

Testing and Evaluation

Software Engineering:

□ *Percent of PQA Test Plan Completed:* 0 %

□ *Percent of PRE ALPHA Test Plan Completed:* ____ %

□ *Percent of PRE BETA Test Plan Completed:* ____ %

Other:

Tested and approved by R&D and Customer Engineering.

Bill of Materials

SOFTWARE:

Component	Version	Description
(Refer to catalog listing)		

MEDIA:

WLI #	Media Description	Media Part Numbers
195-0033-3	1 SSSD Diskette	701-2118P
	or	
195-0033-8	3 Mini-Diskettes	701-8068H 701-8129C 701-8069G

Special Instructions

This release is to be used as an auto-enclosure with the 2200 VP systems. This release should be available to customers on an as-ordered basis only.

Installation Criteria

□ *At next visit*

□ *As soon as possible*

☒ *Customer installed*

□

Approved by

Department Small Systems

6/2/83

COMPANY CONFIDENTIAL

2200 BASIC-2 Single User Operating System

VP

Release 2.6

June 2, 1983
SE Package # 0099

TABLE OF CONTENTS

Section

1.0 RELEASE ABSTRACT

It corrects a problem within Release 2.5 and problems within the previous release of two utilities.

2.0 PRE-REQUISITES

2.1 Hardware

2200 VP

2.2 Software

None

3.0 ALERTS

None

4.0 TESTING/EVAULATION

This release was tested by R&D and by the Technical Operations Group of Customer Engineering. It was found to be 100% acceptable.

5.0 DISTRIBUTION INFORMATION

Release 2.6 is to be used as an auto-enclosure with the 2200 VP systems. This release should be made available to Customer Engineering for use on a per-call situation. It should also be made available to distribution to fill specific customer orders.

6.0 INSTALLATION INSTRUCTIONS

The normal procedure for bringing up the system and installing the software (i.e., '@INSTAL') should be used to install this release.

7.0 MEDIA CONTENTS

195-0033-3	1 SSSD Diskette	701-2118P	2200 System Platter VP BASIC-2 Release 2.6 Release data: 05/13/83
195-0033-8	3 SSSD Mini-Diskettes	701-8068H	2200 System Platter (Diskette 1 of 3) VP BASIC-2 & Utilities Release 2.6 Release date: 05/13/83
		701-8129C	2200 System Platter (Diskette 2 of 3) Utilties (cont'd) Release 2.6 Release date: 05/13/83
		701-8069G	2200 System Platter (Diskette 3 of 3) Diagnostics Release 2.6 Release date: 05/13/83

8.0 RESTRICTIONS AND SPECIAL CONSIDERATIONS

None

9.0 ENHANCEMENTS

None

10.0 PROBLEMS CORRECTED

10.1 System Problems Corrected

(a) The operating system has been corrected so the letters "INIT" do not appear after the "READY" message when RESET is pressed on the 2334DE or 2336DW workstations.

10.2 Utility Problems Corrected

(a) @MOVFIL - Previous releases of this utility would destroy files under the following conditions:

1. The operator specified the same address for both source and destination platters, AND
2. The operator did not specify a destination file name different from the source file name.

The utility has been modified to display an operator prompt in the above situation.

(b) @RECOVER - Previous releases of this utility have not correctly recovered 'all active files' if sectors in the catalogue index (i.e., @INDEX) have contained duplicate file names. This problem was evident with some vendor-supplied software packages.

The utility has been modified to correct for this situation; there has been no functional change to @RECOVER.

11.0 KNOWN ANOMALIES

None

12.0 REFERENCES

None

13.0 ENCLOSURES

None

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

Software Release Notice # 0099

Software Engineering

		ECO No. 28011	*Effective Date* 05/26/83
Description Basic-2 Single User Operating System			*Version* 2.6
Product Line 2200 SVP	*Release Type* Maintenance		*Package Number* 195-2163-5

Purpose of Release Changes have been made to the Operating System and to some of the utilities.

Testing and Evaluation

Software Engineering:

□ *Percent of PQA Test Plan Completed:* ______%

□ *Percent of PRE ALPHA Test Plan Completed:* ______%

□ *Percent of PRE BETA Test Plan Completed:* ______%

Other: Tested and approved by R&D and Customer Engineering.

Bill of Materials

SOFTWARE:

Component	Version	Description
(Refer to catalog listing)		

MEDIA:

WLI #	Media Description	Media Part Numbers
195-2163-5	1 DSDD Diskette	731-0068A

Special Instructions

This release is NOT to be used as an auto-enclosure with the 2200 SVP systems. The MVP Basic-2 Operating System will continue to be the software shipped with these systems.

Installation Criteria

□ *At next visit*

□ *As soon as possible*

□ *Customer installed*

☒ For use on an as-required basis.

Approved by

Department Small Systems

6/2/83

2200 BASIC-2 Single User Operating System

SVP

Release 2.6

June 2, 1983
SE Package # 0099

TABLE OF CONTENTS

Section

1.0 RELEASE ABSTRACT

It corrects a problem within Release 2.5 and problems within the previous release of two utilities.

2.0 PRE-REQUISITES

2.1 Hardware

2200 SVP

2.2 Software

None

3.0 ALERTS

None

4.0 TESTING/EVAULATION

This release was tested by R&D and by the Technical Operations Group of Customer Engineering. It was found to be 100% acceptable.

5.0 DISTRIBUTION INFORMATION

Release 2.6 IS NOT to be used as an automatic enclosure with 2200 SVP systems; the current release of the 2200 MVP BASIC-2 Multi-user Operating System will continue to be shipped with the SVP. This release should be made available to Customer Engineering for use on a per-call situation. It should also be made available to distribution to fill specific customer orders.

6.0 INSTALLATION INSTRUCTIONS

The normal procedure for bringing up the system and installing the software (i.e., '@INSTAL') should be used to install this release.

7.0 MEDIA CONTENTS

195-2163-5	1 DSDD Diskette	731-0068A	2200 System Platter SVP BASIC-2 Release 2.6 Release data: 05/13/83

8.0 RESTRICTIONS AND SPECIAL CONSIDERATIONS

None

9.0 ENHANCEMENTS

None

10.0 PROBLEMS CORRECTED

10.1 System Problems Corrected

(a) The operating system has been corrected so the letters "INIT" do not appear after the "READY" message when RESET is pressed on the 2334DE or 2336DW workstations.

10.2 Utility Problems Corrected

(a) @MOVFIL - Previous releases of this utility would destroy files under the following conditions:

1. The operator specified the same address for both source and destination platters, AND
2. The operator did not specify a destination file name different from the source file name.

The utility has been modified to display an operator prompt in the above situation.

(b) @RECOVER - Previous releases of this utility have not correctly recovered 'all active files' if sectors in the catalogue index (i.e., @INDEX) have contained duplicate file names. This problem was evident with some vendor-supplied software packages.

The utility has been modified to correct for this situation; there has been no functional change to @RECOVER.

11.0 KNOWN ANOMALIES

None

12.0 REFERENCES

None

13.0 ENCLOSURES

None