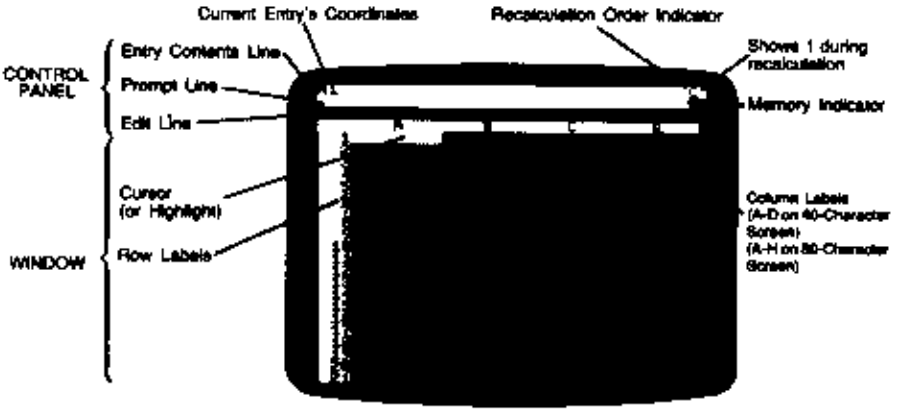
PART III. VISICALC COMMAND REFERENCE

The VisiCalc Screen



The Control Panel

The top three lines of the VisiCalc screen make up the **control panel**. Each line has a specific function:

The **entry contents line** is the top line of the panel. At the left, it will always show the coordinate on which the cursor is currently placed. Next to the cursor it displays the exact content (as it was entered) of that entry position, if there is anything written in it. Any explicit formats (see Part III, the FORMAT Command) that might have been set at the coordinate are also displayed here. At the right side of the entry contents line, the letter **C** or **R** appears, indicating that calculation and recalculation is being done down columns or across rows (see Part III, the GLOBAL Command).

The **prompt line** is the middle line in the control panel. At the left side, it displays the prompts for the command that is currently being used. The number on the right indicates the amount of available memory left in the computer. On the command structure chart at the beginning of each command discussion in the VisiCalc Command Reference, each prompt line is enclosed in a box. Below the box are all the possible actions VisiCalc can take from that point in the command structure. The control panel is said to be "cleared" when there is no prompt on this line. A command can be started only when this line is clear. (The memory indicator on the right will always be visible.) When a command is being used and there is a prompt on this line, pressing RUN/STOP will cancel the command and clear the prompt line.

The **edit line** is the bottom line of the control panel. It displays each character you type or point to with the cursor while using a command. A small rectangle always appears after the last valid character that was typed. Characters that are displayed on the edit line can be erased by backing up the small rectangle with the INST/DEL key and then retyped, if desired. VisiCalc will also, on occasion, use this line to display information which it wants you to confirm or clarify before it carries out a command.

The Window and Sheet

Below the control panel is the VisiCalc **window** which looks upon a portion of the VisiCalc electronic sheet. Across the top of the window there is a border of letters, each of which is a column label. The sheet is divided into 63 columns, labelled A, B, C, . . . BI, BJ, BK. Down the left side of the window there is a border of numbers, which serve as headings for each row. There are 254 possible rows on the VisiCalc sheet.

Entry Positions

The intersection of each column and row **defines an entry position**. A column letter and a row number identifies each entry position, *i.e.*, D17. This identifier is called the entry position **coordinate**.

The Cursor

When VisiCalc is loaded, there is a light bar highlighting entry position A1. This bar is called the **cursor** and sometimes the "highlight." Any command which performs an action on a single entry position will do so in the entry position that is highlighted by the cursor when the command is started. This coordinate remains displayed at the left side of the entry contents line until the command is completed and the cursor moved, although, during the course of some commands the cursor can be moved (an action called **pointing**).

Moving the Cursor

The cursor can be moved to any position on the electronic sheet. The exact keys which cause the cursor to move are discussed in detail in Part I of this manual, in the section entitled "Some Notes on Your Keyboard." In the VisiCalc Command Reference, we continue to use the symbols ↑, ↓, ←, and → to indicate the use of these keys to move the cursor. The cursor will move one entry position in the direction of the arrow each time you press the cursor-moving key.

When the cursor has been moved to the right or bottom edge of the sheet visible through the window, VisiCalc will **scroll** the entire window across or down the sheet, following the cursor so that it is in view at all times. To jump around the sheet quickly, see Part III, the GO TO Command.

If the cursor is "bumping" into any of the four edges of the electronic sheet, the cursor and the coordinate on the entry contents line will flash.

Automatic Repeat

Pressing ↑, ↓, ←, or → and *holding* it down will automatically cause the highlight to move more quickly, in the direction of the arrow. The window will scroll to keep up with the cursor.

Pointing with the Cursor

Whenever you are using a command and can type in a formula (see the VALUE ENTRY Command) or an entry position coordinate, VisiCalc allow you to move the cursor to **point** to the coordinate you want. Check the prompt and edit lines to be sure you have begun the command and VisiCalc is waiting for you to enter a coordinate before you press one of the cursor-moving keys. Then move the cursor to the desired entry position. You will see the coordinate on the edit line change as you move the cursor. If you try to point with the cursor when it is not allowed, VisiCalc may end the command and move the cursor to the next entry position.

Typeahead

At times, you may type faster than VisiCalc reacts to your keystrokes. This is because VisiCalc may be doing any number of things in reaction to the last key you pressed, such as expanding the electronic sheet and recalculating formulas. VisiCalc has a feature called **typeahead** so that it remembers the keys you pressed, no matter how fast you go. It will catch up with you as soon as it can.

Correcting Mistakes

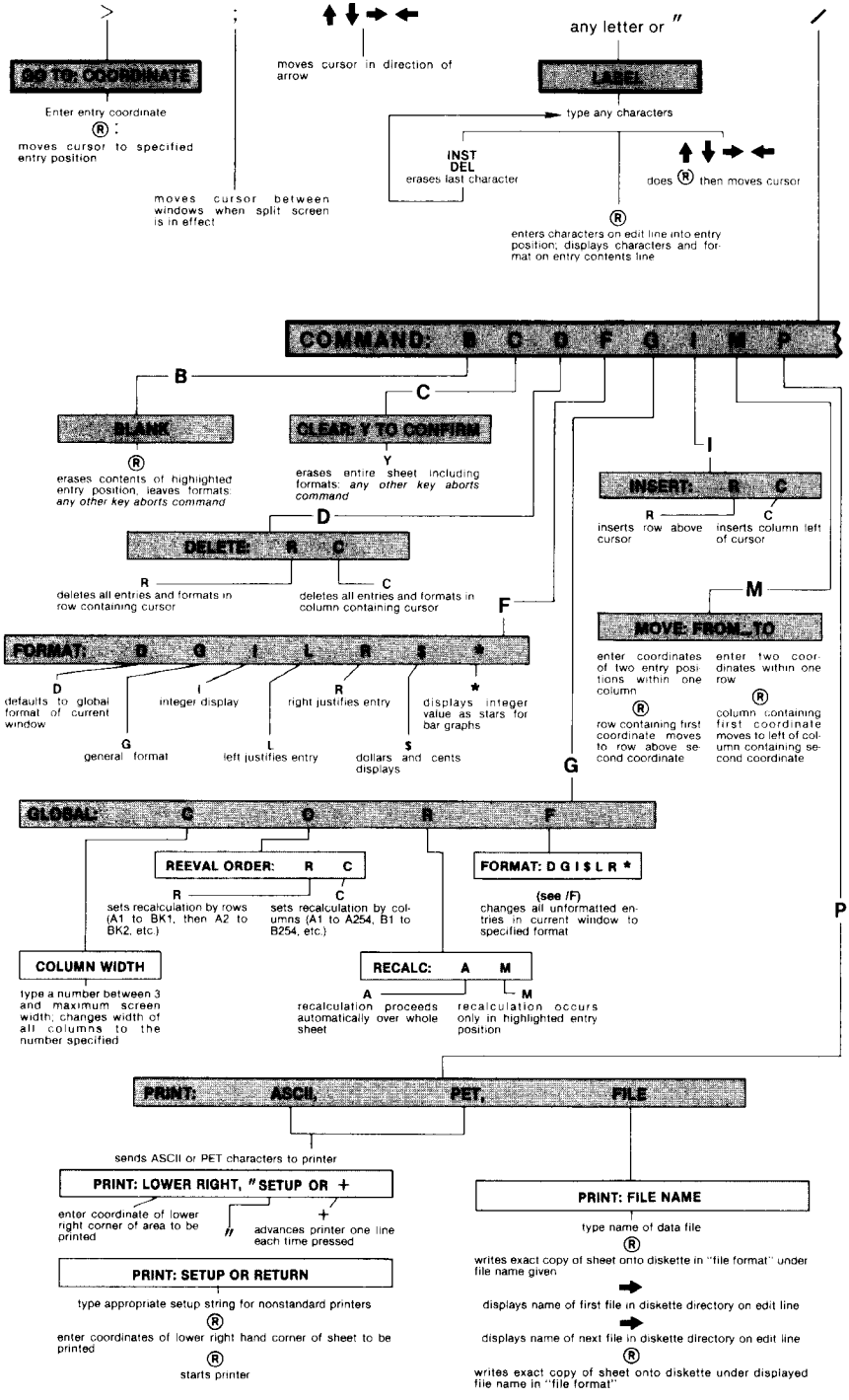
When you have characters on the edit line, you may back up to erase them by using the **INST/DEL** key and then typing in the correct characters. Each time **INST/DEL** is pressed, the small rectangle on the edit line will erase one character. Press it enough times and you'll back completely out of the command and have a clear control panel.

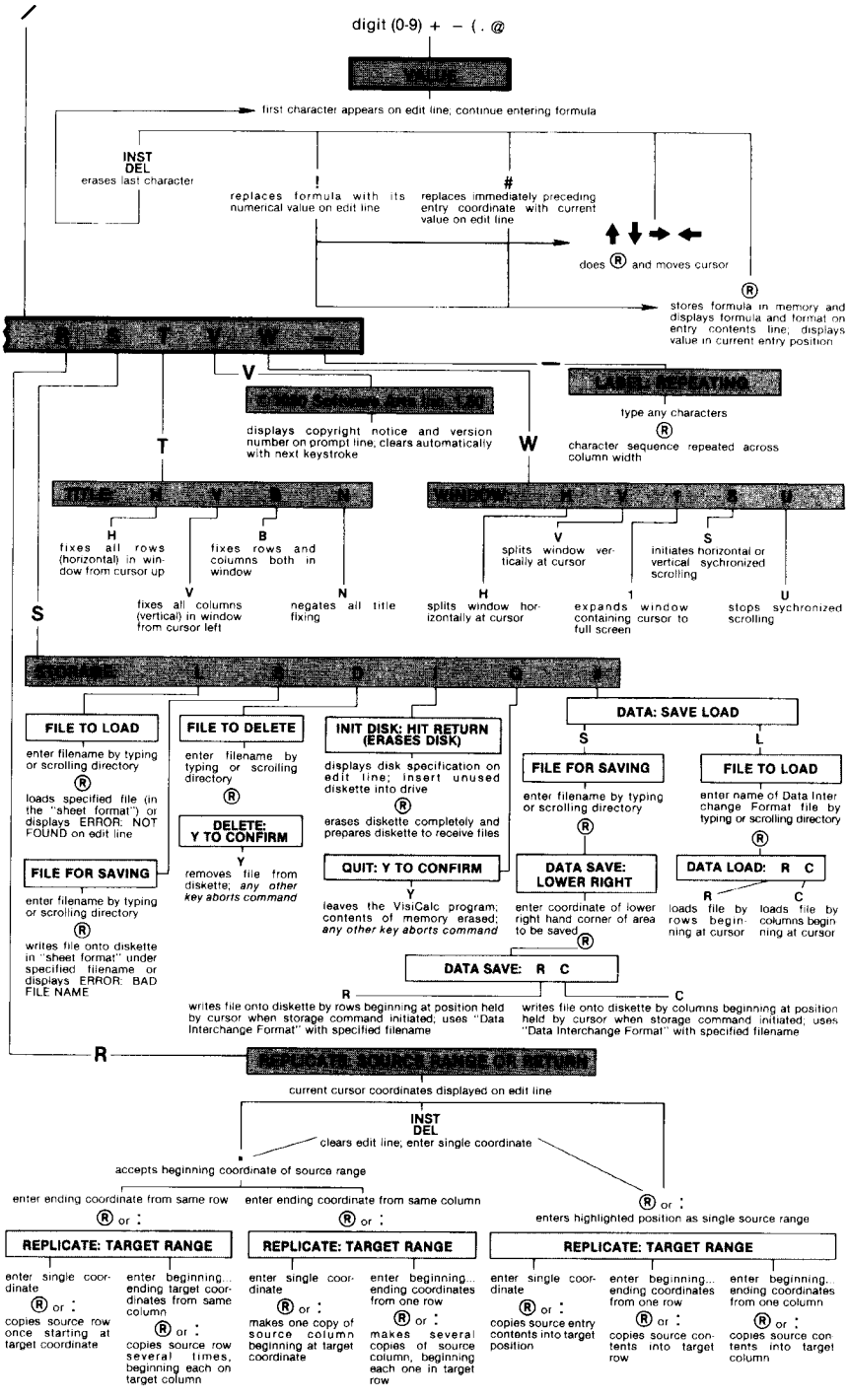
If you start a command and decide you don't want to complete it, press **RUN/STOP** to cancel it and return to a clear control panel.

To change the contents of an entry position, position the cursor at the coordinate, and initiate the desired command. When you finish the command with its appropriate terminator, the old contents will be replaced by the new.

The Memory Indicator

The number at the right side of the prompt line is called the **memory indicator** and tells you how much memory is available as you write on the electronic sheet. There is a complete discussion of the computer's memory and the way VisiCalc uses it in Part II, Lesson Two in the sections entitled "Postscript: Memory and the Electronic Sheet," "How the Sheet is Reconfigured," "The Memory Indicator," and "Dynamic Memory Allocation."


>
GOTO: COORDINATE
Enter entry coordinate
R
:
moves cursor to specified entry position
;
moves cursor between windows when split screen is in effect
moves cursor in direction of arrow
any letter or "
LABEL
type any characters
INST DEL
erases last character
does R then moves cursor
R
enters characters on edit line into entry position; displays characters and format on entry contents line
/
COMMAND: B C D F G I M P
B
BLANK
R
erases contents of highlighted entry position, leaves formats; any other key aborts command
C
CLEAR: Y TO CONFIRM
Y
erases entire sheet including formats; any other key aborts command
D
DELETE: R C
R
deletes all entries and formats in row containing cursor
C
deletes all entries and formats in column containing cursor
I
INSERT: R C
R
inserts row above cursor
C
inserts column left of cursor
M
MOVE: FROM_TO
enter coordinates of two entry positions within one column
R
row containing first coordinate moves to row above second coordinate
enter two coordinates within one row
R
column containing first coordinate moves to left of column containing second coordinate
F
FORMAT: D G I L R $ *
D
defaults to global format of current window
I
integer display
R
right justifies entry
*
displays integer value as stars for bar graphs
G
general format
L
left justifies entry
$
dollars and cents displays
G
GLOBAL: C O R F
REEVAL ORDER: R C
R
sets recalculation by rows (A1 to BK1, then A2 to BK2, etc.)
C
sets recalculation by columns (A1 to A254, B1 to B254, etc.)
FORMAT: D G I $ L R *
(see /F)
changes all unformatted entries in current window to specified format
COLUMN WIDTH
type a number between 3 and maximum screen width; changes width of all columns to the number specified
RECALC: A M
A
recalculation proceeds automatically over whole sheet
M
recalculation occurs only in highlighted entry position
P
PRINT: ASCII, PET, FILE
sends ASCII or PET characters to printer
PRINT: LOWER RIGHT, " SETUP OR +
enter coordinate of lower right corner of area to be printed
"
+
advances printer one line each time pressed
PRINT: SETUP OR RETURN
type appropriate setup string for nonstandard printers
R
enter coordinates of lower right hand corner of sheet to be printed
R
starts printer
PRINT: FILE NAME
type name of data file
R
writes exact copy of sheet onto diskette in "file format" under file name given
displays name of first file in diskette directory on edit line
displays name of next file in diskette directory on edit line
R
writes exact copy of sheet onto diskette under displayed file name in "file format"


digit (0-9) + − (. @
first character appears on edit line; continue entering formula
INST DEL
erases last character
!
replaces formula with its numerical value on edit line
#
replaces immediately preceding entry coordinate with current value on edit line
does ® and moves cursor
® stores formula in memory and displays formula and format on entry contents line; displays value in current entry position
V
displays copyright notice and version number on prompt line; clears automatically with next keystroke
type any characters
®
character sequence repeated across column width
T
W
H fixes all rows (horizontal) in window from cursor up
B fixes rows and columns both in window
V fixes all columns (vertical) in window from cursor left
N negates all title fixing
V splits window vertically at cursor
S initiates horizontal or vertical sychronized scrolling
H splits window horizontally at cursor
1 expands window containing cursor to full screen
U stops sychronized scrolling
S
FILE TO LOAD
enter filename by typing or scrolling directory
®
loads specified file (in the "sheet format") or displays ERROR: NOT FOUND on edit line
FILE TO DELETE
enter filename by typing or scrolling directory
®
DELETE: Y TO CONFIRM
Y
removes file from diskette; any other key aborts command
INIT DISK: HIT RETURN (ERASES DISK)
displays disk specification on edit line; insert unused diskette into drive
®
erases diskette completely and prepares diskette to receive files
QUIT: Y TO CONFIRM
Y
leaves the VisiCalc program; contents of memory erased; any other key aborts command
FILE FOR SAVING
enter filename by typing or scrolling directory
®
writes file onto diskette in "sheet format" under specified filename or displays ERROR: BAD FILE NAME
DATA: SAVE LOAD
S
FILE FOR SAVING
enter filename by typing or scrolling directory
®
DATA SAVE: LOWER RIGHT
enter coordinate of lower right hand corner of area to be saved
®
L
FILE TO LOAD
enter name of Data Inter change Format file by typing or scrolling directory
®
DATA LOAD: R C
R loads file by rows beginning at cursor
C loads file by columns beginning at cursor
DATA SAVE: R C
R writes file onto diskette by rows beginning at position held by cursor when storage command initiated; uses "Data Interchange Format" with specified filename
C writes file onto diskette by columns beginning at position held by cursor when storage command initiated; uses "Data Interchange Format" with specified filename
R
current cursor coordinates displayed on edit line
INST DEL
clears edit line; enter single coordinate
accepts beginning coordinate of source range
enter ending coordinate from same row
® or :
enter ending coordinate from same column
® or :
® or :
enters highlighted position as single source range
REPLICATE: TARGET RANGE
enter single coordinate
® or :
copies source row once starting at target coordinate
enter beginning... ending target coordinates from same column
® or :
copies source row several times, beginning each on target column
REPLICATE: TARGET RANGE
enter single coordinate
® or :
makes one copy of source column beginning at target coordinate
enter beginning... ending coordinates from one row
® or :
makes several copies of source column, beginning each one in target row
REPLICATE: TARGET RANGE
enter single coordinate
® or :
copies source entry contents into target position
enter beginning... ending coordinates from one row
® or :
copies source contents into target row
enter beginning... ending coordinates from one column
® or :
copies source contents into target column

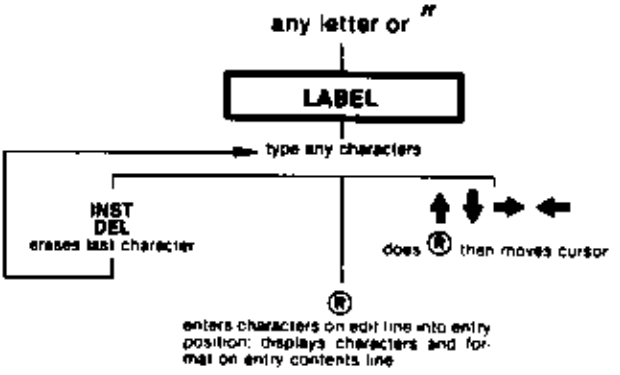
The GO TO Command

With this command, you have a fast way to move the highlight to any position on the sheet. To move the highlight cursor to a new entry position, type >, the coordinates of the entry position to which you want to jump, and ®.

Example

1)	Type >	prompt line:	GO TO: COORDINATE
2)	Type AM205®	The highlight will be moved to column AM, row 205.	

The LABEL ENTRY Command



Any alphanumeric combination of characters can be put into an entry position. The character strings entered as labels are evaluated by VisiCalc as 0.

You begin the label entry command from a cleared control panel. The label will be entered in the entry position on which the cursor is sitting when you begin the label entry command. When the first character of the label is alphabetic, just begin typing the label. VisiCalc immediately recognizes that you are entering a label.

You may want to enter a label such as 2ND MONTH. In such a case where the first character is a digit or one of the arithmetic operators which automatically begin a value entry (see VALUE Command), you must type a quotation mark (") as the first character in the label. The " will not appear as part of the label.

As soon as you have pressed either the " or a letter, the prompt line will display LABEL and the characters of the label will appear in the edit line and in the entry position. Press INST/DEL to back over any mistyped characters and retype them.

When the label is complete, terminate the command with either ®, ↓, →, ↑ or ←. The label will appear in the entry position with the first character at the left side of the column and will have replaced anything that you may have previously placed in that position. You may tell VisiCalc to display the label right-justified within the column with a special format (see the discussion of /FR in Part III, the FORMAT Command). All entries made with the label command are given the entry type L, which appears on the entry contents line as (L).

Example

1) Type >A1	The cursor is in entry position A1.	
2) Type **P**	prompt line: LABEL edit line: P	
3) Type **ERIOD**	entry contents line: prompt line: LABEL edit line: PERIOD	A1 (L) PERIOD

4)	Press ↵	entry contents line:	A1 (L) PERIOD
		prompt line:	clear
		edit line:	clear
	or press ←, ↑, →, or ↓	The label is entered and highlight is on the next entry position.	

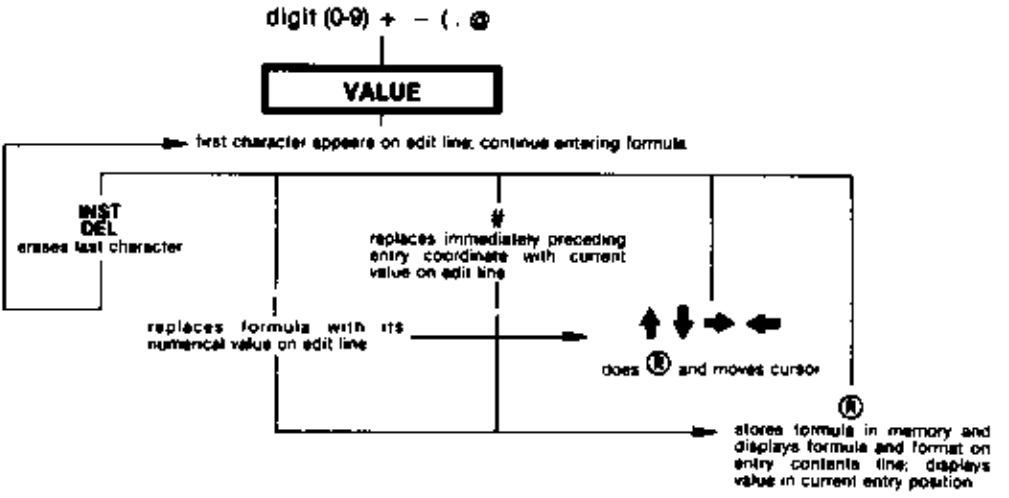
Example

Suppose you want to use a label that looks like a formula.

1)	Type >B1↵		
2)	Type "	prompt line:	LABEL
3)	Type .575*B2	prompt line:	LABEL
		edit line:	.575*B2
4)	Press ↵, ↑, ↓, ← or →	entry contents line:	B1 (L) .575*B2
		prompt line:	clear
		edit line:	clear

The (L) indicates entry type and allows you to distinguish the entry from a value entry.

The VALUE ENTRY Command



A value entry is a formula you type and then enter by pressing ⊕, ⊕, ⊕, ⊕, or ⊕. When a value has been entered, it will replace the previous contents of the entry position. A formula may be a number or an arithmetic expression and may contain entry position coordinates, called **value references** and functions and their arguments. For example, 1000, +B13, 1.5*12/6, +C2*B2 and 1.5+(D2/B3) are all valid formulas.

When a formula contains a value reference, VisiCalc uses the current value in the entry position in its calculation. VisiCalc then displays the calculated value of the formula in the entry position on which the cursor is located when you start the command.

All values are stored with between 11 and 12 significant digits. When a number is displayed in the general format (see the FORMAT Command), VisiCalc will shift between conventional and scientific notation as required to display the calculated value with the greatest precision. In scientific notation, the number 123456789123 becomes 1.235E11. the "E11" means "times 10 to the 11th power." The largest number possible is .999999999999E62. The smallest is 9.999999999E−66. Except in $ format (see the FORMAT Command), non-significant zeros are dropped from the display.

If the column is too narrow to display the number, even in scientific notation, VisiCalc will place as many right angle brackets (>) as it can in the entry position.

An illegal calculation, such as dividing by zero, will result in a value that is displayed as ERROR in the position where the illegal calculation occurs and in all other positions that reference the calculation.

Any of the following keystrokes will initiate the value entry command: a digit + − (. # @

The arithmetic operators are:

+	addition
–	subtraction
/	division
*	multiplication
^	exponentiation

VisiCalc performs the calculations in the order it encounters them, from left to right. To change the order of precedence, use parentheses to indicate "do this first." If there are parentheses within parentheses, VisiCalc will calculate the innermost first. For example, 5+6/2*4 will be evaluated as 22, but 5+((6/2)*4) will evaluate as 17.

The complexity of a formula, *i.e.*, the number of value references, arithmetic operators, parentheses, functions and their arguments, and the amount of memory in your computer determine the maximum length of the formula you may enter. If the formula becomes too complex as you type it, VisiCalc will stop displaying your keystrokes. You will be able to enter everything in the formula up to that point. VisiCalc will not allow you to enter an illegal formula such as one which ends with an arithmetic operator. You must back up with the INST/DEL key to erase the offending character(s).

A formula that does not contain values references may be one such as 1435, or – 14.35 or 5*12+1–60. As soon as you type one of the characters that initiate a value entry (as listed in the first paragraph), VisiCalc immediately places VALUE on the prompt line and displays the character you typed on the edit line. As you continue to type in the elements of the formula, they appear on the edit line. Until you have terminated the command and entered the formula by pressing ® or one of the cursor-moving keys, you can use INST/DEL to modify what you've typed, or press RUN/STOP to leave the value entry command altogether.

When the value entry has been made, VisiCalc displays the calculated value in the entry position on the sheet, but stores the actual formula that was on the edit line when you terminated the entry in its memory. The formula that was used to obtain a value will always be displayed on the entry contents line when you place the cursor on the value. For example, if 5*12+1–60 were the value entry, the entry position would display 1, its calculated value, whereas the entire formula would appear on the entry contents line.

VisiCalc may also be used like a calculator as you are typing in a formula on the edit line. Suppose you wanted to store the result of 5*12+1–60 as the value entry, and not the formula from which it was derived. Simply type the formula on the edit line and then press the exclamation point !. VisiCalc immediately calculates the value. If ® or one of the cursor-moving keys is pressed at this point, the number 1 will appear in the entry position and in the entry contents line. Alternatively, you could continue developing the formula for the value entry.

Visicalc provides a method of duplicating the numerical value in one entry position into another, without having to retype the formula or to replicate it (see the REPLICATE Command). However, with this method, only the calculated value in the entry position is used as the new value entry, and not the original formula. Place the cursor at the position into which the number is to be placed and then initiate the value entry command with +. Then, either point with the

cursor or type in the coordinate of the entry position containing the number you wish to duplicate. The edit line will display the coordinate. Press #, and the *value* of that coordinate will appear on the edit line. Press ® or move the cursor, and the value will be entered into the entry position.

Example

1)	Type /CY	The sheet is cleared and the cursor is at A1.	
2)	Type 1	prompt line:	VALUE
		edit line:	1
3)	Press ®	entry contents line:	A1 (V) 1
		prompt line:	clear
		edit line:	clear
		position A1:	1
4)	Press ♦	entry contents line:	B1
5)	Type 3*34®	entry contents line:	B1 (V) 3*34
		prompt line:	clear
		edit line:	clear
		position B1:	102
6)	Type ♦8*1.5®®	entry contents line:	C1 (V) 12
		position B1:	12
7)	Type >B5®	entry contents line:	B5
8)	Type +	prompt line:	VALUE
		edit line:	+
9)	Press ♦ until the cursor is highlighting B1.	edit line:	+B1
10)	Press #	edit line:	+102
11)	Press ®	entry contents line:	B5 (V) 102
		prompt line:	clear
		edit line:	clear
		position B5:	102

VisiCalc permits entry position coordinates to be used as elements in formulas. These elements, called value references, take on the value in the entry position to which they refer. This value will change whenever you change the contents of the referenced entry position. You cannot just begin typing in a formula that begins with a value reference, since the first character you would type would be a letter and VisiCalc would assume you are typing a label (see the LABEL ENTRY Command). You must initiate the value entry command with a +, or 0+ and then type or point with the cursor to the value reference. If the next keystroke is an arithmetic operator, you may follow it with a number or a value reference. Until you've terminated the value entry command with ®, ♦,♠, ♣, or ♥, you can use the INST/DEL to back up and change the formula on the edit line, or press RUN/STOP to interrupt the command and return to the clear control panel without making an entry.

If you are looking at different parts of the sheet through a split screen (see the WINDOW Command), you can use the ; to jump from one window to another to point to entry position coordinates you wish to use in the formula.

The cursor-moving keys (↑,↓,←,→)will not terminate the value entry command if they were just used to point to a value reference that follows an arithmetic operator on the edit line. For example, if 1+ is on the edit line, and you point with the cursor to A5, the formula on the edit line will be 1+A5. Pressing a cursor-moving key at this point will not terminate the command, but will change the coordinate A5 to correspond to the new cursor position. In this example, ® must be used to enter the formula.

If you press # immediately after a value reference on the edit line, the value of the coordinate will replace the value reference on the edit line. Pressing ! will evaluate the entire formula before it is entered into an entry position, automatically replacing every value reference that is on the edit line with its current value. ® or any of the cursor-moving keys will enter the value of the formula in the entry position. Note that the resulting value will appear in both the highlighted entry position and the entry contents line. The original formula will not be saved.

Example

1)	Type /CY	To clear the sheet. Cursor is at A1.	
2)	Type 1↓	position A1:	1
		Cursor is on A2.	
3)	Type +	prompt line:	VALUE
		edit line:	+
4)	Press ←	edit line:	+A1
5)	Type +1	edit line:	+A1+1
6)	Press ®	entry contents line:	A2 (V) +A1+1
		position A2:	2
7)	Type ↑1+←#	edit line:	1+2
		The value of A2 has replaced the value reference.	
8)	Type !	edit line:	3
		The formula on the edit line has been evaluated.	
9)	Press ®	entry contents line:	A3 (V) 3
		position A3:	3
		Notice that the formula used to derive the value of 3 has not been stored because of the use of !.	

In this example, every time you change the value of A1, you will see the effect of that change in position A2, which contains A1 as a value reference in the formula +A1+1. Try it by placing the cursor on A1 and typing 100®. The value in A2 will be recalculated and replaced with the new value, 101.

Recalculation Order

VisiCalc formulas may contain as many value references as the complexity of the formula will allow. When any value entry is made, including changing an

existing entry, VisiCalc automatically recalculates every value on the sheet. Recalculation always starts in the upper left hand entry position, A1. When first loaded, VisiCalc calculates a value for A1, then A2, then A3, then A4 to the end of column A. Then it recalculates B1, B2, B3, B4 to the end of column B; then C1, C2, C3, C4 and so on. Note the letter C in the upper right corner of the control panel. This indicates that the order of recalculation for the whole sheet is by Column. The global command (see the GLOBAL command) contains an option which lets you change the order of calculation from down columns to across rows. When row calculation is in effect the upper right corner of the control panel will display an R.

Forward and Circular References

Pay peticular attention to the placement of any formulas which contain value references. When in column recalculation, be sure that all referenced entry positions are to the left of the formulas which cite them (or above a formula in the same column). If the sheet is not arranged in this way, the formula containing a value reference will be recalculated before the new value has been placed in the referenced entry position. When recalculation has been completed, the sheet will display the value of the formula as calculated using the *old* value from the referenced entry position. However, the *new* value of the referenced entry position will be displayed in the entry position.

This problem, called forward referencing, is often difficult to diagnose and one is tempted to conclude that VisiCalc has made an arithmetic error. If you suspect your sheet contains a forward reference which is causing a formula to be incorrectly updated, press the ! once. This will force another recalculation of the whole sheet. Watch the suspect formula. If a new value appears, look for forward references. You may choose to redesign your sheet to eliminate all forward references or to use multiple !'s for recalculation. In row calculation, referenced values must be placed in the rows above the formulas which use them or to the left in the same row.

A circular reference is one which cites itself, such as placing the formula, 1+A1, in entry position A1, for example. Each time the sheet is recalculated the value of this formula will change, even if no other changes are made on the sheet. Circular references can be very useful when their results are correctly anticipated. However, they are disasterous when entered by mistake.

VisiCalc's Built-in Functions

Built-in functions are used within value entries to save you the effort of setting up commonly used formulas yourself. Each function begins with @, then comes the "function word," followed by an argument in parentheses. For example, the SUM function might be written:

@SUM function might be written:

@SUM(B1,S2,A4*.23)

This expression would result in the sum of the values found in entry positions B1, S2, and .23 times the value in entry position A4. A built-in function may be placed in any entry position by itself or used as part of a larger expression. The @ may be used as the first character in a value entry (no preceeding+ is necessary). The function word may be shortened to include only enough letters to avoid confusion with other function words. The function words are listed in the table below with the form of their arguments and their definitions.

Function Arguments

All but three of the built-in functions are followed by an argument written in parentheses. The arguments shown as "v" in the chart may be any legal VisiCalc values. The arguments shown as "list" may consist of any combination of values and entry ranges separated by commas. A range is a portion of a row or a column specified by its beginning entry position coordinate, three dots, and its final entry positin coordinate (A3 . . . A17, for example). A range may not be a diagonal across rows or columns. When entering a range, you may type in the coordinate or move the highlight to point at the desired entry position. Entry positions containing labels or blank entries are evaluated as zeros when they are used as value references in function arguments or in formulas.

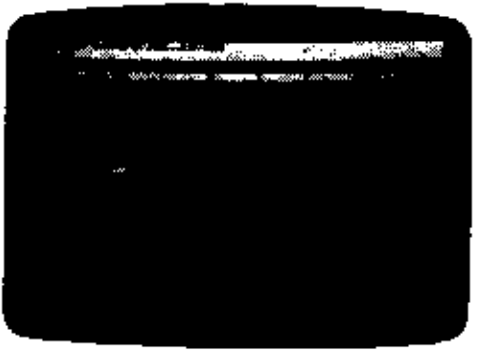
@SUM(list)	adds the values of all entry positions cited in the list.
@MIN(list)	chooses the smallest value in the list.
@MAX(list)	chooses the largest value in the list.
@COUNT(list)	results in the number of non-blank entries in the list.
@AVERAGE(list)	@SUM(list) divided by @COUNT(list)
@ABS(v)	results in the absolute value of v.
@INT(v)	results in the integer portion of v.
@SQRT(v)	results in the square root of v.
@EXP(v)	returns e (2.71828. . .) to the v power.
@LOG10(v)	results in the logarithm (base 10) of v.
@LN(v)	returns the natural log (base e) of a number.
@SIN(v), @COS(v), @TAN(v), @ASIN(v), @ACOS(v), @ATAN(v)	returns the appropriate trigonometric function of the value. Calculations are done in radians.
@NPV(dr,range)	calculates the Net Present Value of the cash flows in the range, discounted at the rate specified by dr (the discount rate expressed as a decimal). The first entry in the range is the cash flow at the end of the first period, the second entry is the cash flow at the end of the second period.
@NA, @ERROR, @PI	These three functions have no argument. @PI is evaluated as 3.1415926536. @NA (not available) is used when a sheet must be set up before the data to be evaluated is available. If entry positions which will later contain data are left blank, they will be evaluated as zero if used as value references in formulas. This will produce the entry ERROR on the sheet wherever zeros appear as denominators and may produce incorrect or misleading values elsewhere. Entering @NA into the blank data positions causes VisiCalc to evaluate all entries which refer to those positions as NA. Without entering the data, you can be assured that

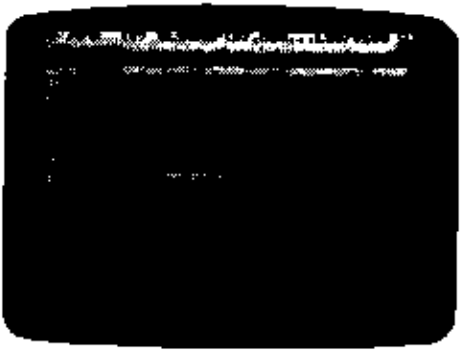
all your formulas are legal in VisiCalc. Any illegal calculations will show up as ERROR. Like @NA, @ERROR is also displayed in the entry position into which it is entered and all positions which refer to it. In addition, the @ERROR function is often generated by VisiCalc in such cases as when a deleted row or column contained entry positions that are value references in formulas.

@LOOKUP(v,range) looks up a numeric value in a table and finds the value that corresponds to it.

The table used by the @LOOKUP function may be a range within a column or a row. The value that is being looked up will be compared to successive values in that table until a value is found that is larger than the value being looked up (or until the end of the table is reached). The entry in the table that is before this entry is the one that VisiCalc will consider as the "match" for the value being looked up. If the table is in a column, VisiCalc returns the value of the entry that is immediately to the right of the "matching" entry as the value of the function. If the table is in a row, VisiCalc returns the value immediately below the "matching" entry as the value of the function. If the first value in the table is greater than the value being looked up, then the value of the function will be NA.

An example of the use of the @LOOKUP function would be the following invoice form. The total amount of the goods purchased is looked up in a table and the found value is used as a discount percentage. Finally, the dollar amount of the discount is calculated, and the total amount of the invoice is found. In this example, the order of calculation is by rows (see the GLOBAL Command, /GO).

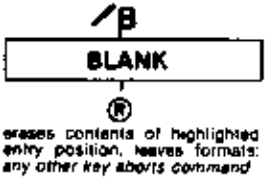




The formula at entry position B9 is @LOOKUP(D7,A18 . . . E18)*100 (by multiplying by 100, we make it a percentage). Although the lookup range is technically a forward reference (see the VALUE ENTRY Command), it makes no difference in this case because the values in that range are constants. The format of that entry position is /FG (see the FORMAT Command). The formula at D9 is +D7*B9/100. The lookup table has been placed in positions A18 through E18.

If the total of the invoiced items was $347.52, then the discount percent would be 10% (the value that corresponds to $250.00). A total of $3000.00 would have a discount of 25%, the same as for $1000.00.

The BLANK Command



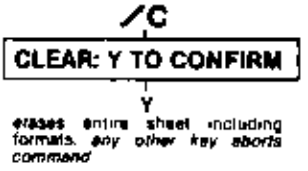
Blank removes only the label or value in the entry position on which the highlight cursor is located, but leaves the existing format setting for that position.

Use this command to erase an entry after you have pressed ⊕ or moved to another entry position.

Example

1)	Move the highlight over the entry position to be erased.	
2)	Type /	prompt line: COMMAND: BCDFGIMPRSTVW
3)	Type B	prompt line: BLANK
4)	Press ⊕	The control panel will clear and the highlighted entry postion will be blank.
	or press ↑,↓,←, or →	The control panel will clear, the original entry position will be blank, and the highlight will be on the next entry position.

The CLEAR Command



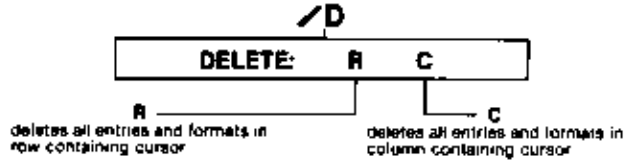
Clear sets all entry positions to blank, resets formats, titles, windows, and other information about the sheet to the initial specifications set by VisiCalc when you first load it into the computer. The entry position highlight is returned to entry position A1.

Use this command to start with an empty VisiCalc sheet. Before using the clear command, be sure you have saved the sheet (see the STORAGE Command) if you do not want to lose the information you had written on it. All information erased with the clear command is irretrievable.

Example

1)	Type /C	prompt line: CLEAR: Y TO CONFIRM
2)	Press Y	The screen will darken for a few seconds, then display the copyright and version notices. Your next keystroke will clear the prompt line.

The DELETE Command

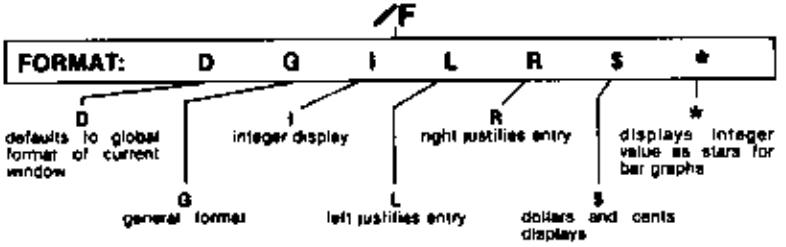


VisiCalc allows you to remove the entries from an entire column or row. Use the delete command with care: *You cannot recover the information which was recorded in a deleted row or column.* The insert commands (see the INSERT Command) can recreate the row or column space, but not the actual entries. The delete command will delete the information in the row or the column which contains the cursor when you start the command and then move all the other rows or columns up to fill in the slack.

Visicalc also automatically looks up all value references in formulas and changes them to correspond to the new coordinates resulting when the rows or columns are moved up. It then recalculates the whole sheet. If you have deleted an entry which is used in a formula (see the VALUE ENTRY command), all entry positions affected will contain the message **ERROR**.

To use this command, first place the highlight in the row or column you want to delete. Then type **/D**. The prompt line reads DELETE: R C. Type **R** to delete the row in which the highlight is located or type **C** to delete the column. Visicalc immediately deletes the information and moves all other rows or columns up to fill in the slack.

The FORMAT Command



The format command adds a set of display conditions to the highlighted entry on the VisiCalc sheet. The original entry contents (LABEL or VALUE) remain unchanged in the computer's memory. They are always used in calculations and are displayed completely on the entry contents line. On the sheet, however, the entry will appear as formatted by this command. If the entry is moved to another position on the sheet (see the REPLICATE Command, the MOVE Command, and the INSERT Command), printed on paper (see the PRINT Command), or stored on diskette (see the STORAGE Command) its format stays with it.

Once an entry position has been formatted, a format indicator will appear on the entry contents line between the current entry coordinates and entry contents. Erasing the contents of the entry position (see the BLANK Command) does not remove explicit formats, but clearing the sheet (see the CLEAR Command) does.

To format a row or a column before making entries, format the first entry and then use the replicate command (the REPLICATE Command) to copy the format into the rest of the row or column.

It is not possible to change the number of character spaces in an individual entry position. To change the number of characters in every column thoughout the sheet, use the global command for changing column width (see the GLOBAL Command).

To see how the format options affect the display, set up your sheet as instructed in the example below. All the examples in this section use this sheet for the format examples.

Example

1) Type /CY — This clears the sheet, positions the cursor at A1, and resets all display characteristics.

2) Type **LABEL ENTRY**
"1.23456789"99.999

You now have one label and two values on your sheet. Any explicit format settings were removed when the sheet was cleared. You have one window (see the WINDOW Command), no titles (see the TITLE Command), your columns contain nine character spaces, which is the general column width (see the GLOBAL Command), and all entry positions display the general format (see the GLOBAL Command and the discussion below). These same display characteristics are set each time VisiCalc is loaded into your computer and whenever the clear command is used.

/FD is the **Default** display format:

An individual entry position format is changed to whatever format has been previously set with the global command (see the GLOBAL Command). If no global setting is in effect, the default will be the general format, described in **/FG** below.

Example

1) Type >C1↵

2) Type **/F$**

The highlighted entry (C1) changes to dollars and cents format; B1 is unchanged. The entry contents line displays the /F setting.

entry contents line: C1 /F$ (V) 99.999
position C1: 100.00

3) Type **/FD**

entry contents line: C1 (V) 99.999
position C1: 99.999
The default setting is general, as set by /CY.

4) Type **/GFI**

All values in the window which have *not* received an individual format are displayed as integers.

position B1: 1
position C1: 100

5) Type **/FG**

entry contents line: C1 /FG (V) 99.999
position B1: remains in integer format
position C1: 99.999
The setting on C1 is now general, as shown on the entry contents line.

6) Type **/FD**

entry contents line: C1 (V) 99.999
position B1: unchanged
position C1: 100
The setting on C1 has been removed and is no longer indicated on the entry contents line. However, since there is a global integer setting in effect, position C1 is displayed in integer format instead of general format.

/FG sets the entry position to the **General** format:

a) Labels begin in the leftmost space (left justified) and are cut off wherever the column width ends.

b) Values are moved as far to the right as possible (right justified) with a leading blank character in the leftmost position in the column. Decimal or scientific notation is selected to display largest number of significant digits.

Example

1) Type >A1®	position A1 is now in general format for labels entry contents line: A1 (L) LABEL ENTRY
2) Press ↓	position B1: 1 A global integer setting is still in effect from the previous example.
3) Type /FG	position B1: 1.234568 General format for numbers with few significant digits.
4) Type 1234567891234567890®	entry contents line: 1234567891230000000 position B1: 1235E17 General format for numbers with many significant digits. VisiCalc selects scientific notation when this permits a larger number of significant digits to be displayed within the current column width.

/F I will set a value to the **integer** format:
A value set with this command is displayed rounded to the nearest whole number.

Example

1) Type 1.2®	entry contents line: B1 (V) 1.2 position B1: 1.2
2) Type /F I	entry contents line: B1 /FI (V) 1.2 position B1 1 Entry contents are rounded to the nearest integer.

/FL Left justifies the entry position contents:
All labels begin in the leftmost character space; all values are moved to the left, preceded by one leading blank. The integer format for values cannot be used when a value is left justified. This effect of this command is visible only in entry positions where the entry has fewer characters than the current column width.

Example

Continuing with the example for integer format (/FI):

1) Type /F L	entry contents line: B1 /FL (V) 1.2 position B1: 1.2 The value in B1 has been moved to the left. Notice that the integer format setting has been replaced with the left justify setting in the entry contents line.

/FR Right justifies the entry position contents:
The last character of a label or a value falls in the last character space of the entry position. This command affects only the display of entries on the sheet which have fewer characters than the current column width.

Example

1) Type /FR

entry contents line: B1 /FR (V) 1.2
position B1: 1.2
The value in B1 has been moved to the right.

/F$ displays values in **Dollars and Cents**:
All values are rounded to two decimal places. No $ is displayed in the entry position. Trailing zeros are shown. There is no effect on labels.

Example

1) Type /F$

entry contents line: B1 /F$ (V) 1.2
position B1: 1.20

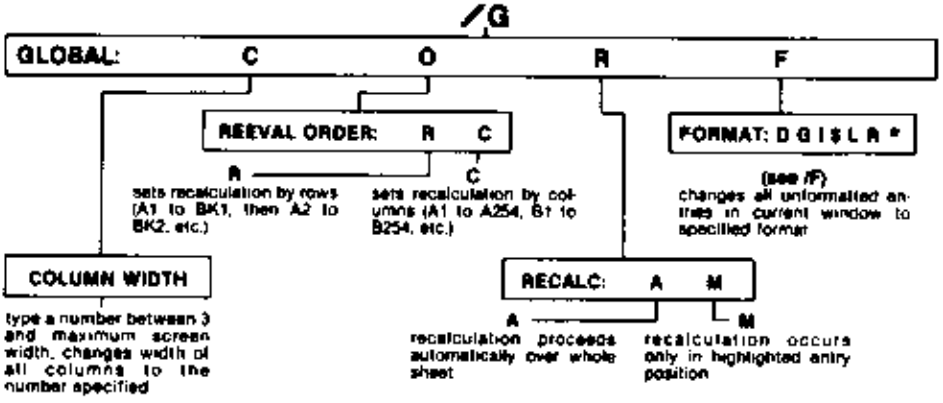
/F* displays **bar graph** format:
All values are truncated to an integer. That integer is replaced in the entry position by an equal number of stars. If the number of stars is larger than the width of the entry position, extras are ignored.

Example

1) Type /F*

entry contents line: B1 /F* (V) 1.2
position B1: *

The GLOBAL Command



The four global command options affect the window display and the order in which rows and columns are calculated.

/G C (Global Column) allows you to change the width of all columns in a window to any number of characters between 3 and the maximum width of your screen. Column widths may not be set individually. VisiCalc will display as many whole columns of a given width as possible on the screen. When changing to a single, wide column in a window, any vertical title areas previously set will be automatically removed (see the TITLE Command).

When displaying numbers on the sheet, VisiCalc may change the format from the way you originally entered it on the edit line. Numbers will be rounded off in the window display when necessary to fit into the column width. VisiCalc will use scientific notation if this will permit more significant digits to be shown at the current column width than regular decimal notation. If the column is too narrow to display the integer portion of a value in either decimal or scientific notation, VisiCalc will display as many > symbols as will fit in the column width in the entry position with a leading blank.

Regardless of how the values appear on the screen, the numbers stored in the computer's memory will remain unchanged and will be used in all calculations. The number, as you originally entered it on the edit line, will be displayed on the entry contents line whenever the cursor is over its entry position. Pressing # when the cursor is over the entry and the prompt line is clear will enter the entire value, regardless of column width, onto the edit line.

Example

1)	Type /CY	This clears the sheet and resets the column width to nine characters.
2)	Type in position A1: **THIS LINE IS TOO LONG.**↵	entry contents line: THIS LINE IS TOO LONG.
3)	Type /G	prompt line: GLOBAL: C O R F

4)	Type C	prompt line:	WIDTH:
5)	Type 18↵	entry contents line:	THIS LINE IS TOO LONG.
		position A1:	THIS LINE IS TOO L
6)	Type /GC24↵	entry contents line:	THIS LINE IS TOO LONG.
		position A1:	THIS LINE IS TOO LONG.
		position B1:	not displayed on 40 character screens.

Example

1)	Type /CY		
2)	Type 123456789↵	entry contents line:	A1 (V) 123456789
		position A1:	1.2346E8
		Note the leading blank, scientific notation, and final digit rounded up.	
3)	Type ↵1.2345678950↵	entry contents line:	B1 (V) 1.23456789
		position B1:	1.234568
		Note the leading blank and the final digit rounded up.	
4)	Type /GC3↵	entry contents line:	B1 (V) 1.23456789
		position A1:	>>
		position B1:	1.
		The number in A1 is too large to express with two characters, which is the space available for digits when the column width is 3, so VisiCalc displays >>. The number in B1 is rounded as shown.	

/GO (Global Order of recalculation) allows you to set the Order of Calculation to Columns or Rows. In columnwise calculation and recalculation, entry position A1 is evaluated, then A2, A3 to the bottom of the sheet, then B1, B2, B3, to the bottom of the sheet in this column, then C1, C2, and so on. In row calculation, A1 is first, then B1, C1, and on to the right hand end of the sheet, then A2, B2, C2 to the end of the sheet. This order of calculation is indicated on the control panel by the letter C or R in the upper right hand corner of the entry contents line. When you first load VisiCalc, the calculation order is by column.

If VisiCalc appears to evaluate formulas incorrectly, you have placed formulas in entry positions so that they are calculated before the value references that they contain. Order of calculation and recalculation has been discussed at length in The VALUE ENTRY Command in Part III, and also in Part II, Lesson Three in the section entitled "The Order of Recalculation."

The following example illustrates the use of value references set up for column calculation, and shows what happens when they are calculated in a rowwise fashion.

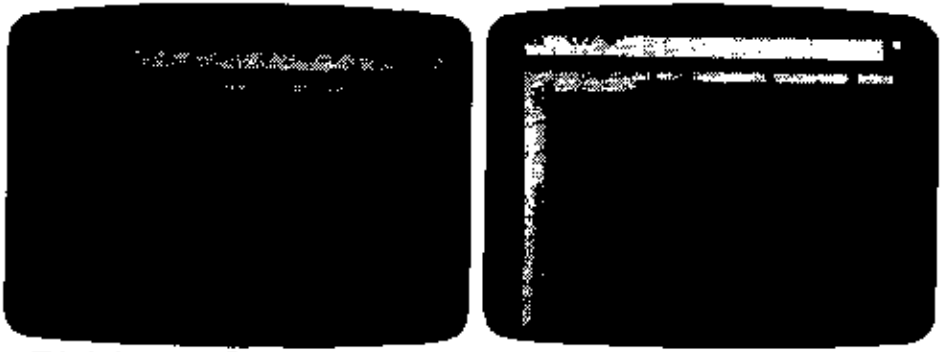
Example

1) Type /CY — This clears the sheet and sets column calculation.
2) Type 1 ↵
3) Type +A1 ↵
4) Type >B1 ↵
5) Type +A2 ↵
6) Type +B1 ↵ — Your screen should look like this:

7) Type >A1 ↵2 ↵ — The value at A1 is changed to 2, and the other values are recalculated. Your screen should look like this:

8) Type /GOR — The order of calculation indicator changes to R.

9) Type 3® — Your screen looks like this:



This is incorrect. B1 should be the same as A2, yet it displays a different value because B1 was recalculated *before* A2 was recalculated. In such instances, you can force a recalculation by pressing ! when the prompt line is blank. Press it now and the screen shows the correct values as in the photo below.

A similar example that works correctly for row order of calculation, but not for column order is this:

Example

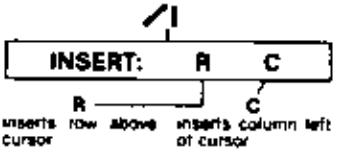
1) Type /CY — The sheet is cleared and calculation is set to column order.
2) Type /GOR — To change to row order of calculation.
3) Type 1® + A1®
4) Type >A2®
5) Type + B1® + A2®

/GR initiates Automatic or Manual Recalculation. When first loaded, VisiCalc is set to recalculate all values as soon as any change is made in an entry position to which a formula may refer. Changes ripple rapidly across your sheet whenever you change a value. Sometimes, such as when you are typing in a

column or row of figures, you would like to turn off recalculation completely to eliminate the pause that occurs when VisiCalc is recalculating. To turn it off, type /GRM, which stands for GLOBAL COMMAND:RECALCULATION MANUAL. Under manual recalculation, only the highlighted formula will be recalculated automatically. The whole sheet is updated only when you type !. There is no cue on the control panel to indicate whether you are in manual or automatic recalculation but the status should be evident from watching the behavior of the sheet. To resume automatic recalculation, type /GRA.

/GF (Global Format) allows you to assign a format setting for all entry positions in the window which have not been individually formatted. The global format command uses the same formatting options as are available for individual entry positions. See the FORMAT Command for an explanation of these options. If you have split your screen into two windows (see the WINDOW Command), you may use a different global format for each window. When you save your sheet with /SS all global formats are also saved (see the STORAGE Command).

The INSERT Command



While developing a VisiCalc sheet, you may find that you need to insert additional rows or columns. The insert command gives you this ability. After you type /I, VisiCalc expects you to enter **R** if you want a row inserted, or a **C** to insert a column. It is not possible to insert a single entry position — you may only insert an entire row or column.

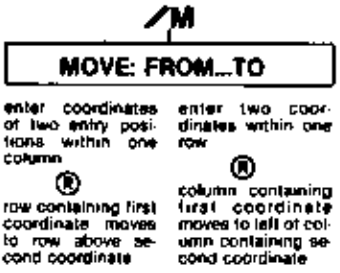
/IR inserts a new, blank row in the row containing the cursor when you give the command. All rows that were at or below the cursor are moved down one row to make room for the insertion. VisiCalc then changes all value references in formulas to reflect the new entry position coordinates in the rows that were moved. For example, if a formula contains the coordinate C2, and a row is inserted at row 2, VisiCalc will change the coordinate to C3. The cursor will remain in its old position in the new, blank row.

After typing /IR

/IC inserts a new, blank column in the column in which the cursor is located when the command is given. All columns at and to the right of the cursor are moved one column to the right to make room. VisiCalc changes all value references in formulas to reflect the new entry position coordinates for the columns that were moved. The cursor remains in its original position in the new, blank column.

After typing /IC

The MOVE Command

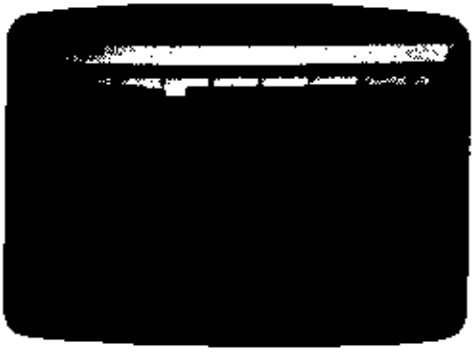


It is sometimes desirable to change the position of a row or a column. The move command lets you do just that. You indicate the row or column you want moved by placing the cursor in the row or column, then type a period, and then place the cursor in the row or column to which you want to move. VisiCalc will automatically change all value references in formulas to reflect the new coordinates which result when the rows or columns are moved. When you move a row or a column, be careful that you have not moved formulas containing value references to a position in which they are calculated before those value references. Order of calculation and recalculation is discussed at length in the VALUE ENTRY Command in Part III and in Part II, Lesson Three in the sections entitled "The Order of Recalculation" and "Forward and Circular References."

The simplest way to move a row or a column is to first place the cursor on the row or column you wish to move. Then type /M. The coordinate of the highlighted entry position will appear on the edit line. Press . and then use the cursor-moving keys to place the cursor on the row that is just below the row to which you are moving or on the column just to the right of the desired new location. In other words, the column or row will appear just to the left or just above the cursor. When moving rows, the cursor may be in any position (column) within the row when you initiate the command; however, when you indicate the new location, you must use the same column coordinate. For instance, moving C10 to C5 is valid, but moving C10 to D5 is not. The same is true for moving columns: the row coordinate must be the same for the column being moved and its new location; *i.e.*, moving A2 to D2 is valid, but moving A2 to D1 is not. Terminate the command with ®. When a row or column is moved, all intervening rows and columns are moved up to fill in the place vacated by the moved row or column. VisiCalc then changes all value references to reflect the new positions.

Example

1) Set up your sheet as in the photo:

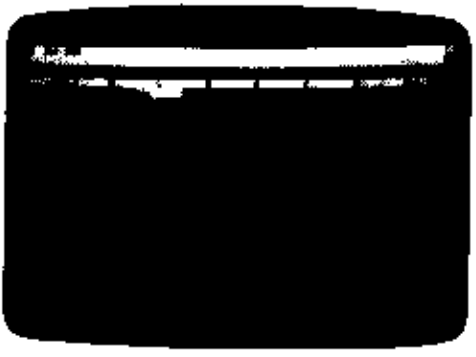


2) Type /M — prompt line: MOVE: FROM. . .TO; edit line: C1

3) Type . — edit line: C1. . .
Row 1 is going to be moved.

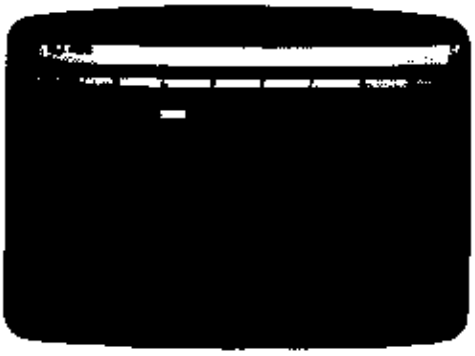
4) Press ♦♦♦ — edit line: C1. . .C4
Row 1 is going to move to row 3.

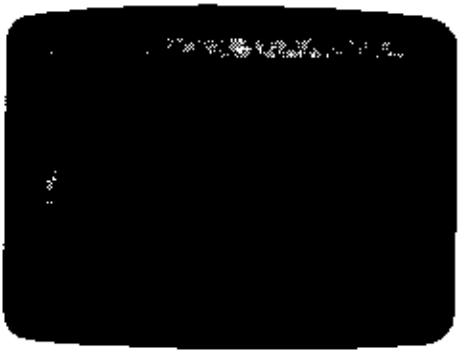
5) Press ® — The sheet should look like the photo below:



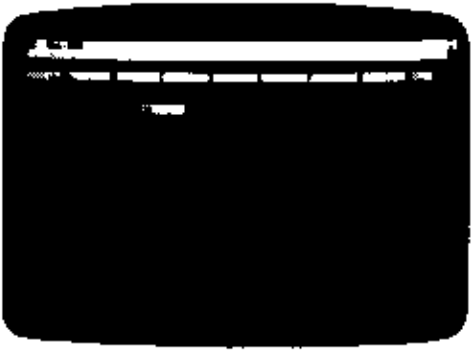
Example

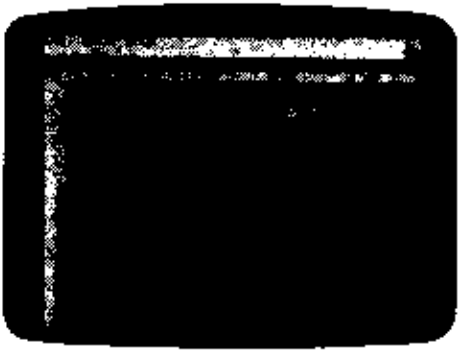
1) Set up your sheet as in this photo:





2)	Type /M	prompt line: FROM...TO	MOVE:
		edit line:	C3
3)	Type .	edit line:	C3...
		Row 3 will be moved.	
4)	Press ◆	edit line:	C3..C2
		Row 3 will move to row 2.	
5)	Press ↵	Row 3 has moved to row 2.	

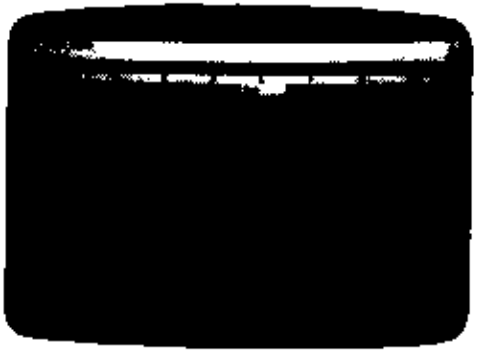




It is also possible to type in the coordinate of the row or column to be moved as well as the coordinate of the new location. Use the INST/DEL key to erase any unwanted coordinate and replace it, either by typing or pointing to the coordinate with the cursor.

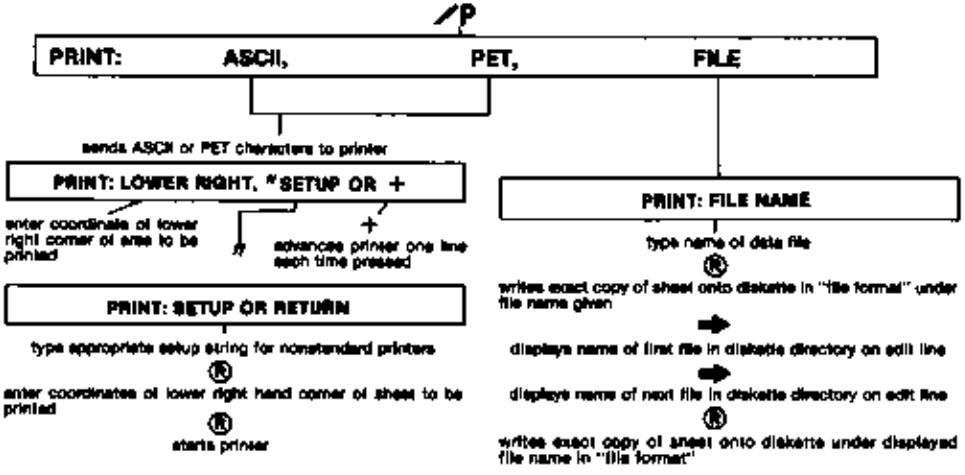
Example

1) Set up your sheet as in the photo below:



2)	Type /M	prompt line: FROM...TO	MOVE:
		edit line:	E1
3)	Type INST/DEL	edit line:	blank
4)	Press ◆◆◆◆	edit line:	A1
		Column A will be moved.	
5)	Press .	edit line:	A1...
6)	Type E1	edit line:	A1...E1
		Column A will be moved to column D. Although the cursor is on E1 at the end of step 5), pressing ® at that point will not enter E1 on the edit line.	
7)	Now press ®	The contents of column A are now in D. The columns that were previously B, C, and D have been moved to A, B, and C.	

The PRINT Command



The print command sends a copy of your VisiCalc sheet to your printer, to your diskette, or to any other addressable device. This discussion will cover the commands for the Commodore Printer Models 2022 and 2023, the NEC Spinwriter, and the CBM Model 2040 dual floppy disk. It is possible to use the print command to communicate with other printers and ASCII devices such as those often connected to telephone modems. However, the exact commands needed depend on the characteristics of the equipment you are using, how you have it connected to your computer, and how you wish your copy to be formatted. Since it is beyond the scope of this manual to describe the command sequences for all possible hardware combinations, consult the reference manuals for your particular equipment or see your dealer for assistance.

Since your VisiCalc sheet is often wider than the printer, VisiCalc permits you to copy your sheet in strips. All formats (explicit and global) are taken from the window in which the highlight rests and window divisions (see the WINDOW Command) as well as row and column borders are ignored during printing.

Position the cursor over the entry position that you wish to be printed first (in the upper left-hand position on the printout). The cursor's position determines the top row and leftmost column that will be printed. Titles that were fixed (see the TITLE Command) will be ignored, so you must insert extra rows or columns for title areas and replicate the titles (see the INSERT Command and the REPLICATE Command) if you want the titles to appear on the printout. When your sheet is arranged and the cursor is over the first entry position to be printed, follow the steps below:

1) Type /P. The prompt line will display: PRINT: ASCII, PET, FILE

If you wish to print on a NEC Spinwriter (which is an ASCII printer), type A . This tells VisiCalc to send out the right kind of characters (ASCII). The prompt line will now display: PRINT: LOWER RIGHT, "SETUP, OR + . Proceed to step 4.

IMPORTANT NOTICE

A new feature has been added to this version of the VisiCalc program for the Commodore computers. Add this information to page 121 of your manual—The PRINT Command.

The new feature has been added for users with ASCII printers. It allows you to control the line feed automatically sent by the VisiCalc program. When /PA is typed, a new prompt will appear:

PRINT: LOWER RIGHT, "SETUP, +, −, &.

The last characters, − and &, let you issue commands to control the line feed. Note that when you press one of these characters, no change takes place on the VisiCalc screen. Further, you don't have to press (R) after pressing one of the characters. The VisiCalc program is set to send a line feed automatically with each carriage return.

− Pressing − turns off the line feed automatically generated by the VisiCalc program. The VisiCalc program will not send line feeds again until you reboot or until you restore the line feed function with & (see below). The − should only be typed once. If your VisiCalc printout is double-spaced, pressing − before defining LOWER RIGHT will cause all subsequent printouts to single space.

& Pressing & restores a line feed with each carriage return. This will remain in effect until you reboot the VisiCalc program or until you press − (see above).

If you wish to print on a Commodore Printer, type P . This sends out Commodore characters. The prompt line will display: PRINT: LOWER RIGHT, "SET UP, OR +.

You will not need the "SET UP option with the NEC or Commodore printers. This option allows you to type characters that are sent directly to an output device. These characters are the "set up string" that allow you to send to printers other that those discussed here. You would type " and then the characters, ending with a ® (which is not transmitted).

4) If you wish to advance the paper, type +. The paper will advance one line each time you press +. You may advance the paper as far as you want. Use this option only with printers.

5) Now enter the coordinate of the lower right hand corner you wish to print. You may do this by scrolling the window over and placing the cursor over the desired coordinate, then pressing ®. Alternatively, you may just type it in and press ®. This will tell VisiCalc how many columns and rows are to be printed. The width of your printout (the number of columns of a particular width) is limited by the number of characters your printer can put on one line, not by the size of your screen.

6) Now press ® and the printer will copy the area of the sheet that you defined with the coordinates. If the columns you are printing have "wrapped around" on the paper, you tried to occupy more columns than your printer's width could accommodate. Do another printing, using the first column that wrapped around as the upper left hand coordinate and repeat the printing procedure. Tape or glue your printouts together to create a full-width copy of your VisiCalc sheet.

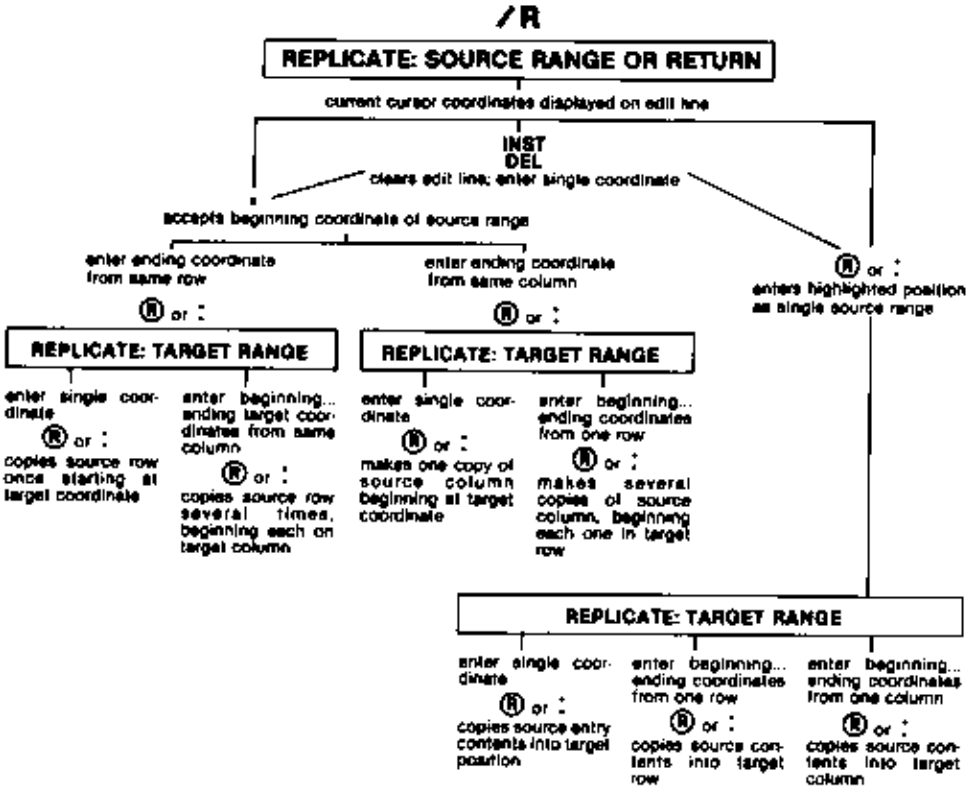
To copy your display into a "print format" file on a diskette so that another program can read it, start with the cursor positioned on the first entry that you want printed in the file, then:

1) Type /P. The prompt line will display: PRINT: ASCII, PET, FILE.

2) Type F in response and enter a file name as asked by the prompt: PRINT: FILE NAME. The **file name** must begin with a letter or number, followed by up to 11 other letters and numbers (including –, underscore, slash, space, and $). End the file name with the characters, **.PRF**, for (pr)int (f)ile. For example, MYFILE–1.PRF is a valid file name. To write the file on a diskette in the left hand disk drive, begin the file name with the characters, **1:** . There is a complete discussion of file names and file specifications in Part III, The STORAGE Command.

3) Press ® and when the prompt line LOWER RIGHT, "SET UP, OR + appears, enter the coordinate of the lower right hand corner you wish to print. You may do this by scrolling the window over and placing the cursor over the desired coordinate, then pressing ®. Alternatively, you may just type it in and press ®. The file will be written onto diskette in print file format. Note that VisiCalc cannot read or reload files in this format. Use /PF only when you intend to write your own BASIC or other language programs for use with VisiCalc.

The REPLICATE Command



The replicate command allows you to copy the contents of any entry position, including labels, values (numbers, expressions, and formulas), formats (the FORMAT Command), and blank entries (the BLANK Command). You may copy a single entry from one place to another, copy a single entry position into a row or a column, or you may reproduce a single row or column as many times as you like. However, the replicate command does not permit you to copy a row into a column, or to copy a matrix (several rows or columns at once). Complex copying operations must be done with a series of replications or with the /S#S option (see the STORAGE Command).

To use the replicate command, you must supply two (and sometimes three) sets of information:

1. the source range
2. the target range
3. indication of the relationship of a value reference (see the VALUE ENTRY Command) to the new position

A range may be a single entry position, as well as a whole or partial row or column. Ranges are specified by typing or by pointing with the cursor to the desired beginning entry coordinate, then typing a period, then typing or pointing to the ending coordinate. A range may not cross the sheet diagonally. A complete source and target range specification will appear on the edit line in the following pattern:

beginning source entry position coordinate . . . ending source entry position coordinate: first target entry position coordinate . . . ending entry position coordinate

If the beginning and ending coordinates in your *source range* are identical, you will copy one entry only. If they are different but fall in the same column, you will copy that section of that column. If they are different and fall in a row, you will copy that section of that row.

The coordinates you put in the target range tell VisiCalc the starting position for each copy to be made with the replicate command. If you want one copy of an entry position, a row, or a column, your target range should contain one coordinate (F9, for example). If you specify two different coordinates for your target range (F9 . . . F15, for example), you will get multiple copies of your source.

To copy from one entry position to another, give one coordinate as the source range and one as the target range:

Example

1) Type >A1®	This places the cursor over the entry position you want to copy. If A1 is blank, type **100®**
2) Type /R	prompt line: REPLICATE:SOURCE RANGE OR RETURN edit line: A1 By starting the command with the cursor on A1, VisiCalc automatically entered A1 to begin the source range.
3) Press ®	prompt line: REPLICATE: TARGET RANGE edit line: A1 . . . A1: You have told VisiCalc to begin copying the contents from entry position A1 and to end with A1. A1 is the "source range" consisting of one entry position.
4) Type D1	prompt line: REPLICATE: TARGET RANGE edit line: D1 This identifies entry position D1 as the start of the "target range," that is, the entry position to which the contents of A1 will be copied.

5) Press ⏎ The replication is completed. The value 100 is now in entry position D1 as well as A1, the highlight is still at A1, and the control panel has been cleared for a new command.

To create a row by making several copies of one entry, give one coordinate as the source range and two coordinates within a single row as the target range:

Example

Repeat steps 1 through 4 in the example above. Change step 5 to:

5) Type **.M1⏎** The contents of entry position A1 will now appear in entry positions D1, E1, F1 through M1.

This procedure is especially useful for setting up display formats (see the FORMAT Command) before entering a large group of numbers. Assume, for instance, that a column will contain sales figures and therefore should always display numbers rounded to two decimal places. Place the cursor on a blank entry position and type **/F$** (see the FORMAT Command) to attach the "dollars and cents" format to the position. Then replicate that entry position into the positions in the column that you want to have the dollars and cents format, using the procedure in the example above. The entry positions will not change in appearance if there are no values in them at this point. However, when you begin entering numbers into them, they will all be displayed with two decimal places.

To copy a column from one position on the sheet to another, give the top and bottom entry position coordinates of the column as the source range: For instance, A1...A32. For the target range, give only the coordinate of the top entry position of the new column.

To set up your sheet for this example, enter the numbers 1 through 10 in positions A1 through A10. You need not clear the sheet because VisiCalc will write over any old entries. (See the VALUE ENTRY Command.)

Example

1) Type **/R**

prompt line:
REPLICATE: SOURCE RANGE OR RETURN

edit line:
current cursor position coordinate.

VisiCalc will put the coordinate of the entry position to which the cursor is pointing on the edit line. If this is not the coordinate you want to begin your source range, press **INST/DEL** to remove it from the edit line.

2) Press **INST/DEL** To erase first coordinate (this is an illustration for this example and is not necessary when the coordinate is the one you want to begin the source range).

3) Type **A1** edit line: **A1**

4) Type .	edit line:	A1 . . .
5) Type A10	edit line:	A1 . . . A10
6) Press ®	prompt line:	REPLICATE:TARGET RANGE
	edit line:	A1 . . . A10:
7) Type C4	edit line:	A1 . . . A10: C4
8) press ®	The contents from your source range (A1 to A10) are now in your target (C4 to C13). Note that you did not need to type the ending coordinate of the target range. The cursor has been returned to the position it occupied before you typed /R.	

To make several copies of a column, enter the top and bottom coordinates of the column as the source range and the beginning and ending coordinates of a row as the target range. Each copy of the source column will "begin" in the target row.

Do the example above again, only this time change steps 7 and 8 to:

7) Type D1.K1 — edit line: A1. . .A10:D1. . .K1
In the target range, D1 will be the top of the first new column, E1 the top of the second new column, and so on, ending with K1 as the top of the last new column. The target range must be adjacent coordinates in one row or VisiCalc will only copy the column once and stop.

8) Type ® — The results will look like this:

If necessary, scroll your window over to K1 to see all of the effects of these few keystrokes on your sheet.

To copy a row from one position to another, specify the beginning and ending coordinates in that row as your source range (A1 . . . C1). Then give the beginning coordinate only for the row in which you want the copy to appear (A5). VisiCalc will automatically interpret this target coordinate as the first entry position in a row and will fill in the correct ending position.

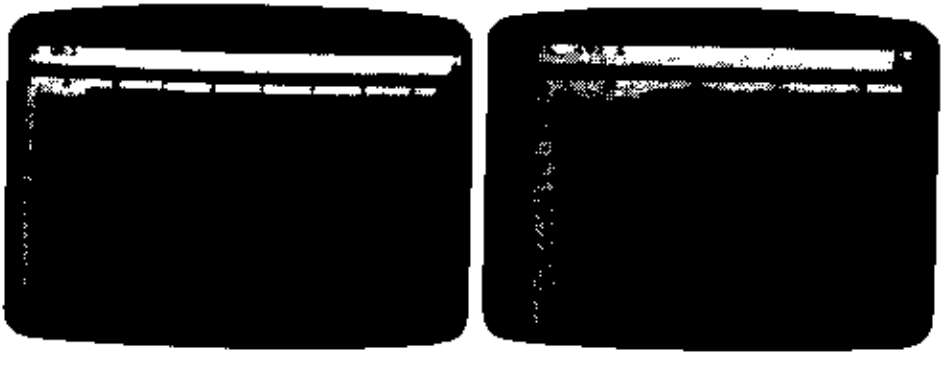
To set up for this example, clear the screen with /CY and enter the numbers 1 through 5 across the top row on your sheet (positions A1,B1,C1,D1,E1):

Example

1) Type >A1
2) Type /RLC1® edit line: A1 . . . C1:
3) Type A5
4) Press ®

The result should look like this:

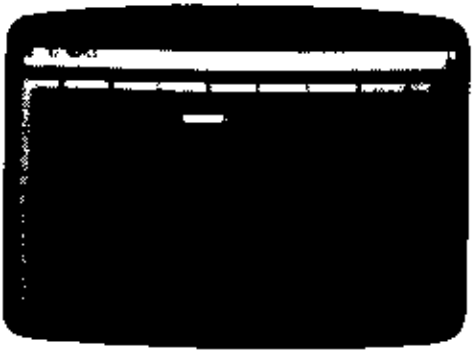
Give two coordinates for the target range (A5 . . . A10, for example) only if you want these results:

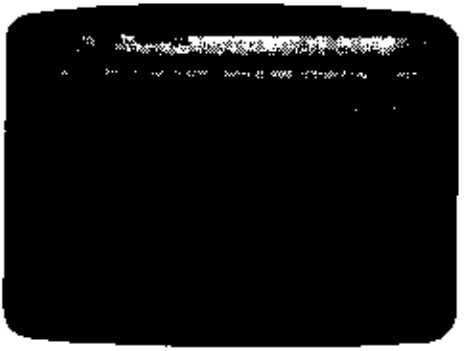


Replicating Value References

When you replicate a formula which contains entry position coordinates (value references), you must tell VisiCalc whether or not to copy each value reference exactly as it appears in the source range, or to change that value reference as

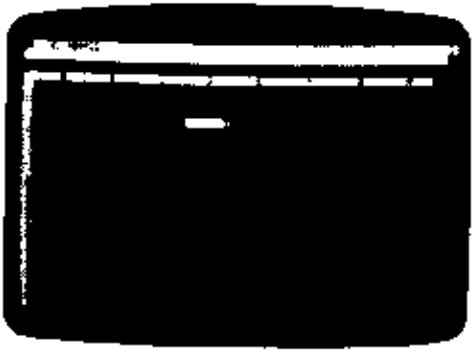
it is copied. The change will replace the original value reference with the one that falls in the position that is relative to the location of the copied formula. Therefore, the replicated value reference will be to the replicated formula as the original value reference is to the original formula. The following examples will illustrate this relationship:

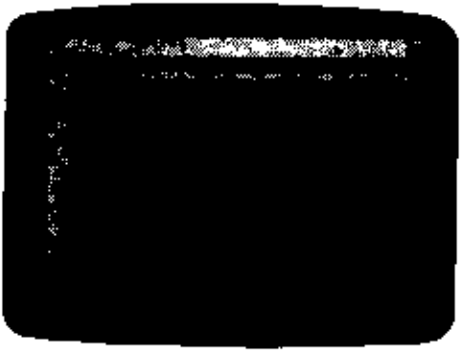




You can see from the entry contents line in the photo above that the formula in position D3 contains value references to position B3 and C1. B3 is in the same row as the highlighted formula and two columns left. C1 is two rows up and one column over from D3.

Look at the photo below.



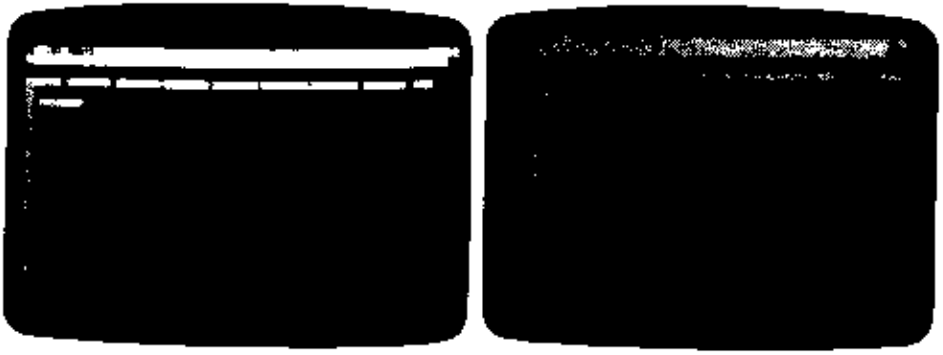


The formula in D3 has been replicated into position D4 but it has also been changed. The formula in D4 is now B4+C1. The value reference B4 is in the same row and two columns left of the new formula in D4, i.e. it is in the same *relative position*. The value reference C1 is exactly the same in the formula at D3 and the formula at D4. In other words, it has not been changed to maintain the same position relative to the placement of the formula.

When a source range that is being replicated contains value references, VisiCalc places each value reference on the edit line with a highlight on it. The prompt line instructs you to type **R** if the value reference is to be relative, or **N** if it is not to be changed in the new formula. After this procedure has been completed for each value reference, VisiCalc will finish the replication.

Example

1) Type /CY — To clear the sheet.
2) Set up your sheet as in this photo:



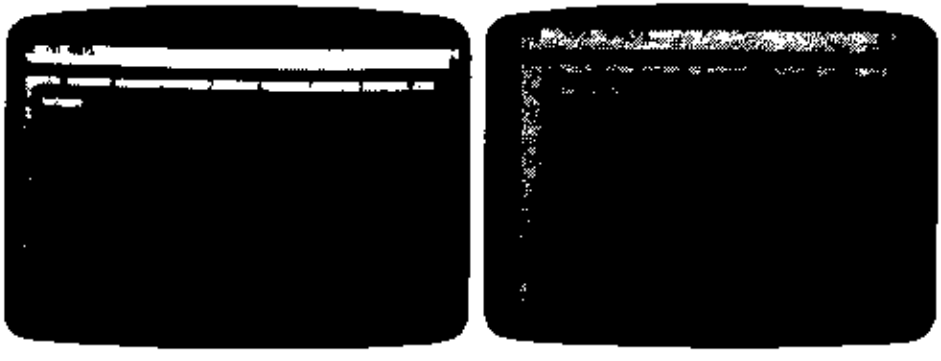
3) Type /R↵ — prompt line: REPLICATE: TARGET RANGE
edit line: A2 . . . A2:

4) Press ↓. — prompt line: REPLICATE: TARGET RANGE
edit line: A2 . . . A2:A3 . . .

5) Press ↓↓↓↓ — prompt line:
REPLICATE: R = RELATIVE, N = NO CHANGE
edit line: A2:A3 . . . A5:A1

The value reference A1 in the formula being replicated is highlighted by the small rectangle. Replication actually takes place one entry position at a time. The edit line indicates that the first operation is copying from A2 into A3 and VisiCalc is waiting to be told whether to interpret the value reference, A1, as relative or unchanging.

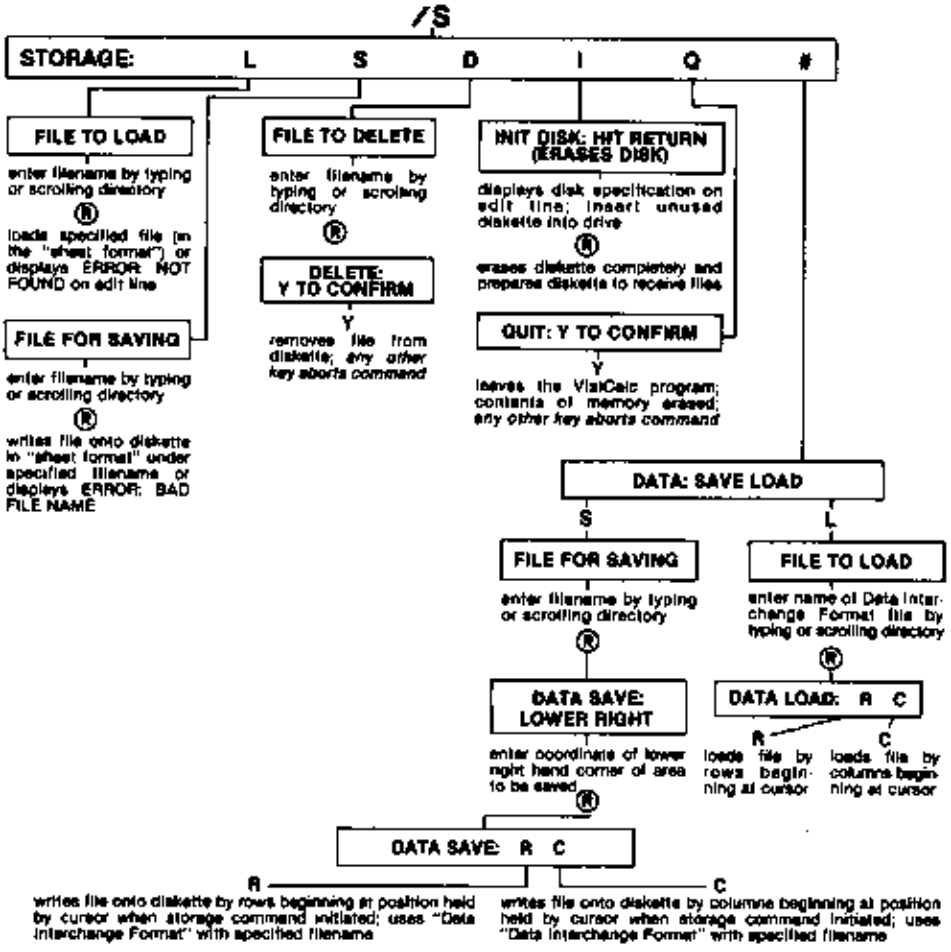
6) Type N — Your sheet will look like this:



It is possible to replicate a formula into a position in which proper relative value references cannot be assigned. Turn back to the two screen photos at the beginning of this section ("Replicating Value References"). If, for example, you copy the formula at D3 into position A3, then the relaive position of the value reference B3 will be located off the sheet to the left! It will always be evaluated as zero.

In Part II, Lesson Three of this manual, there are more examples of the use of the replicate feaure. Also, you must be careful not to introduce forward or circular references or incorrect calculation order into your sheet when you replicate formulas. These topics are thoroughly discussed in Part II, Lesson Three in the sections entitled "The Order of Recalculation " and "Forward and Circular References" and also in Part III, the VALUE ENTRY Command.

The STORAGE Command



The storage command lets you to save copies of your current VisiCalc sheet on diskette, load saved sheets back into the computer, initialize new diskettes for use with VisiCalc, and gives you a way to leave VisiCalc and return to the computer's READY. prompt so you can use your computer for something else.

/SL loads back into the computer's memory a sheet which was saved with the /SS command, which is discussed below. Only files which were saved in the VisiCalc file format with the /SS command (we instruct you to append the extension **.VC** to the file name to help identify these files) should be loaded with the /SL command.

When the file has loaded, the sheet will appear in the window exactly as it was when you gave the /SS command to save it. VisiCalc will not clear the sheet that is in use when the /SL command is given, but will write the loaded sheet

over it. Wherever an entry position has contents on both the current and the loaded sheet, the entry position contents of the just-loaded sheet will replace the previous contents. Blank entry positions in the loaded sheet will not erase the contents of corresponding entry positions in the old sheet. This over-writing characteristic gives you the ability to combine sheets by loading previously saved sheets over one another.

Clear the sheet by typing /CY (see the CLEAR Command) before loading a sheet when you want only the contents of the saved sheet to be placed in the computer's memory.

Insert the diskette containing the file to be loaded and follow the steps below. You may insert the diskette into either disk drive as VisiCalc will look in both drives for a diskette, and then search through the file directory on each one.

1) Type /S	prompt line: STORAGE: L S D I Q #
2) Type L	prompt line: FILE TO LOAD
3) Type the file name used when the file was saved with /SS. Or press ⇻	edit line: The file name of the first file that was saved in the VisiCalc file format with the /SS command will appear. Each time you press ⇻, VisiCalc will examine the directory and place the next qualifying file name it finds on the edit line. Continue pressing ⇻ to "scroll" through the directory until the name of the file you wish to load appears on the edit line.
4) Press ®	VisiCalc will load the file with the name that was on the edit line. While the file is loading, characters will flash at the left side of the edit line. These are the same keystrokes you had originally entered to create the sheet that is now being loaded. When loading is finished, the sheet will appear in the window, looking just like it did when you saved it.

The directory scrolling described in step 3, above, not only puts the existing file names on the edit line. When a file name is on the edit line, you may change the name by backing up the small rectangle with INST/DEL and then typing the characters you wish. This feature might be useful when, for example, you have forgotten a file name. As you scroll the directory a file name appears that jogs your memory as to the name of the file you want to load. Say the name on the edit line is PROJ–JAN.VC and you remember that the file you want is called PROJ–AUG.VC. Press the INST/DEL key until JAN.VC has been deleted from the edit line, and type AUG.VC, then press ® and PROJ–AUG.VC will be loaded (if it is on a diskette that is currently in one of the disk drives).

/SS is the storage save command and will save an electronic sheet, just as you have created it, on a diskette in the VisiCalc format so that it can be loaded back into the computer's memory with the /SL command. Before you can use a

diskette with the storage save command, the diskette must be initialized (you may use VisiCalc's /SI command, explained below in the section on /SI). All formulas, labels, title settings, explicit and global formats and window settings as well as the position of the cursor will be saved and in force when the sheet is loaded.

Each sheet is saved, or recorded, on a diskette under a name you specify, called the **file name**. The file name is recorded in the diskette's "directory." Each file name is unique, so that if you save a sheet with a file name already in the directory, the file with that file name will be replaced by the most recent file saved with the name. There is enough room on a diskette to hold many electronic sheets. Should the disk become full while VisiCalc is recording a sheet into a file, the message ERROR:DISK FULL will appear on the edit line. VisiCalc will have saved all that it could under the file name you gave so you will want to delete that incomplete file from the full disk (see /SD below) and then save the sheet on another, less full, initialized diskette.

The file name can be up to 16 characters long (including spaces). The first character of the name must be a letter or a number. The rest of the name may use letters, numbers, space, dash, underscore, period, dollar sign and slash. We recommend that you append the characters .VC to every file name used for a sheet saved with the /SS command. This suffix will allow you differentiate files you saved in the VisiCalc format from any other files you may have on your diskette. If you choose to use the .VC suffix, the rest of the file name may be only 13 characters long.

When you load VisiCalc to start a session, the computer will automatically assume that the diskette on which you want to save a file is located in drive 0. Drive 0 is said to be the "default" drive. If you want to save the file on a diskette in drive 1, you must preface the file name you enter with the characters 1:. For example, 1:MYFILE.VC would be saved on the diskette in drive 1. The 1: is the "drive specification." Once you've given a drive specification, that drive number becomes the default drive and VisiCalc will automatically save to the diskette in the last-specified drive number.

To save a file, do the steps listed below. The cursor may be anywhere on the sheet when the /SS command is started.

1) Type /S — prompt line: STORAGE: L S O I Q #

2) Type S — prompt line: FILE FOR SAVING

3) Type the disk drive number, if necessary, and the file name.
Or you can press ▸ and VisiCalc will place the first file name it finds in the directory on the edit line. Each time ▸ is pressed, the next file name in the directory will appear, as VisiCalc scrolls through the directory. You may change the file name by pressing INST/DEL to back up the small rectangle and then typing in the new characters. When you have the desired name do step 4.

4) Press ® — The disk drive will whir as the electronic sheet is saved on the diskette under the file name you specified on the edit line in step 3.

The option, in step 3, of letting VisiCalc put file names from the directory on the edit line can be very useful. For instance, if you wanted to keep a record of different versions of a VisiCalc sheet, you might identify the versions by changing only the last few characters of each file name. You could scroll the directory until the last version's file name appeared, then backspace with INST/DEL to delete the version identifier, then type in the new version characters. For example, using this method, the file name BUDGET–4/16.VC could readily be edited to be BUDGET–5/1.VC. It is also a convenient way to examine what file names are in the diskette directory so that you do not unintentionally use an existing file name for the file being saved.

/SD will delete from the diskette the file whose name appears on the edit line.

1) Type **/SD** prompt line: FILE TO DELETE
2) Type the file name, preceded by the drive number, if necessary.
   Or you may press * and VisiCalc will place the first name it finds in the diskette's directory on the edit line. Each time you press *, VisiCalc will examine the directory and place the next file name it finds on the edit line. Continue pressing * to "scroll" through the directory until the name of the file you wish to delete appears on the edit line.
3) Press ® prompt line: DELETE: Y TO CONFIRM
   Any other key will cancel the command.
4) Type Y VisiCalc will delete the file from the diskette.

/SI initializes a new, blank diskette or a previously used, but unwanted diskette. During the initialization process, VisiCalc scans the diskette and writes a series of position identifiers on it. Any contents on a used diskette will be erased during the process. A diskette must be initialized with this command before it can be used to save VisiCalc sheets.

When you load VisiCalc to start a session, the computer will automatically assume that the diskette to be initialized is located in drive 0. Drive 0 is said to be the "default" drive. If you want to initialize the diskette in drive 1, you must tell VisiCalc the disk drive number. The drive number is called the "drive specification." Each time you specify a drive number other than the default, the drive specified becomes the default drive and VisiCalc will then assume it is to use the diskette located in that disk drive.

The command to initialize a diskette may be given at any time when the control panel is clear. The electronic sheet currently in the computer's memory will not be affected by the command. The instructions below assume the default drive is 0.

1) Insert a blank or unwanted diskette into a drive
2) Type **/SI** prompt line:
   INIT DISK: HIT RETURN (ERASES DISK)
   edit line: 0:VC,XX,8
   The first character, the 0, is the drive specification.

3) Press ® The diskette in drive 0 will be initialized.

If the diskette to be initialized is in another drive, press INST/DEL to erase everything on the edit line. Then type the drive number followed by:

:VC,XX,8®

The diskette in the drive specified will be initialized leaving the current VisiCalc sheet unchanged. The drive you specified is now the default drive.

/SQ will let you leave the VisiCalc program and return to Commodore's READY prompt (the same one that's on the screen when you first turn on the computer). Any VisiCalc sheet currently in the computer's memory will be lost. You will have to reload the VisiCalc program (see step 3 below) if you wish to continue using VisiCalc.

1) Type /SQ prompt line: QUIT: Y TO CONFIRM
2) Type Y The computer leaves the VisiCalc program. Any other key cancels the command.
3) To begin VisiCalc again, hold down **SHIFT** and press **RUN/STOP** for the 8032 computer. For the 2001 series, type: **LOAD** "*",8® and then type **RUN**®.

Saving Files in the Data Interchange Format Files saved with the command /S#S are recorded on the diskette in the Data Interchange Format (DIF). This format affords a way for other programs, such as those written in BASIC or FORTRAN, to use the data that is on the sheet. The data that was saved in files with the /S#S command can be loaded back onto a VisiCalc sheet with the /S#L (load a Data Interchange Format file) command.

/S#S saves a rectangular area of the sheet, which you define, in a file in the Data Interchange Format. This command saves labels, blanks, and *calculated* values as they appear in the entry positions on the sheet. The formulas from which the values were derived are not saved (see the VALUE ENTRY Command for discussion of formulas and values). The DIF allows for two orientations of the data. You can specify to VisiCalc which orientation you want by pressing R or C at the appropriate time. For data that you are only going to use with VisiCalc, always save the data with the R specification. As an aid to remembering this, pressing ® is same as typing R. For a complete description of the use of the Data Interchange Format, see the "Programmer's Guide to the Data Interchange Format" inserted in the binder holding this manual. To use the command:

1) Position the cursor at the entry position that is in the upper left hand corner of the rectangular area you want to save.
2) Type /S# prompt line: DATA: SAVE LOAD
3) Type S prompt line: FILE FOR SAVING

4) Type the filename	The filename must follow the same rules outlined in the discussion of /SS, above, except that you should append the suffix .DIF to files saved with this command. Do not use the .VC suffix, which is for use with files saved with /SS.
Or press ◆	To look at the file names already in the directory. Continue pressing ◆ until the desired file name appears on the edit line. You may change the name on the edit line by pressing INST/DEL and typing other characters.
5) Press ®	prompt line: DATA SAVE: LOWER RIGHT
6) Type or point with the cursor to the entry position in the lower right corner of the rectangular area to be saved. You may save only one column or one row if you want.	
7) Press ®	prompt line: DATA SAVE: R, C OR RETURN
8) Press ®	The rectangular area of the sheet defined by the upper left and lower right cursor positions will be saved.

/S#L will load back the data you saved with the /S#S command into any position on the sheet you indicate.

1) Position the cursor on the entry position in the upper left corner of the area to be filled by the data being loaded.

2) Type /S#L	prompt line: FILE TO LOAD
3) Type the desired file name	The file named should be one saved in the Data Interchange Format.
Or press ◆ to scroll through the directory as described in the /SL command discussion, above, until the desired file name appears.	
4) Press ®	prompt line: DATA LOAD: R, C OR RETURN

5) Press ® or R	The selected file is read into the computer's memory and placed on the VisiCalc sheet starting at the current cursor position.

If you would like the data loaded transposed — what would have been loaded across the rows is loaded down the columns, and *vice versa* — press C at step 5.

More on File Names:

Each sheet is stored in a file. The file must be given a unique file name. The entire file specifications for the Commodore computers are:

The drive number (0: or 1:)
The file name (A on 12 character file name beginning with a letter or number. It may include the numbers, letters, period, slash, underscore, question mark, space and dash.
The file suffix (.VC for VisiCalc format, .DIF for the Data Interchange Format, .PRF for the PRint File format)
A valid IEEE bus address (see the Commodore manual for the particular equipment)
The device type (TP for the Commodore printer, TD for the Commodore disk drive, TA for any ASCII device)
The character code (CA for ASCII characters, CP for Commodore characters)
The file type (S for Sequential, P for Program, U for User.)

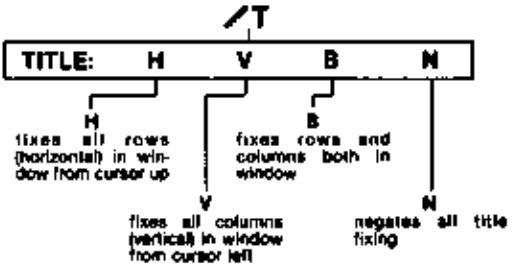
All specifications except the file name and the file format may be left off and VisiCalc will fill in the following default specifications:

–Save (write) to disk drive 0 (0:)
–Load (read) from both disk drives
–IEEE address 8,3 (,8,3)
–Commodore disk drives (,TD)
–Commodore characters (,CP)
–Sequential file type (,S)

The specifications (except for the drive number) must follow the file name and be separated from the file name and each other by commas. They can be appended to the file name in any order. Once you have changed the defaults, the last specifications used become the new defaults.

For instance, instead of saving a sheet to diskette with the /SS command, you might direct the data to be "saved" on a Commodore printer by typing ,TP instead of a file name. Then you can see how the data is actually saved in a diskette file. This is useful for verifying entry position contents such as the formulas stored in entry positions.

The TITLE Command



Most VisiCalc sheets are considerably larger than the screen display window. To see all the entries on the sheet, you must move the window away from the top and left edges, so that any row and column titles you may have entered move out of sight.

The title commands allow you to fix titles in place on the screen, so that they remain in view as you scroll the window about the sheet. Begin the command from a cleared control panel by typing /T. The prompt line on the control panel will display TITLES: H V B N. The possible keystrokes are:

H	To fix horizontal titles.
V	To fix vertical titles.
B	To fix both horizontal and vertical titles.
N	No titles (to "unfix" titles).

Which columns and/or rows are to be fixed is determined by the position of the cursor when you initiate the command. All rows at and above the highlight are fixed by H. All columns at and to the left of the highlight are fixed by V. B fixes all rows above and columns to the left of the highlight.

VisiCalc terminates the command automatically and clears the control panel immediately. There is no change on the sheet, but the effect of title fixing becomes apparent when you begin scrolling the window away from the top and left borders.

You cannot use the arrow keys to move the highlight to an entry position that is within a fixed title area. The highlight will flash when it bumps into the fixed titles. You jump the title barrier by using > (see the GO TO Command) and the coordinates of an entry position within the fixed title area; for example, >A1⏎.

In certain circumstances, VisiCalc will automatically unfix vertical titles you have set. It does this when in order to follow your commands, the vertical titles must not be set. For instance, if you scrolled the window so that column B was at the left edge, then fixed vertical titles and then at some point gave the command to go to a coordinate in column A (see the GO TO Command), VisiCalc would unfix the vertical titles. In order to obey the command to go to column A, the window had to be scrolled to the left to bring A into view.If you have fixed vertical titles, and then you expand the column width (see the GLOBAL Command) so that only one column can be displayed on the screen at one time, VisiCalc will automatically "unfix" the vertical title settings. When you return to a narrower column width, you will have to refix the vertical titles.

The VERSION Command

/V

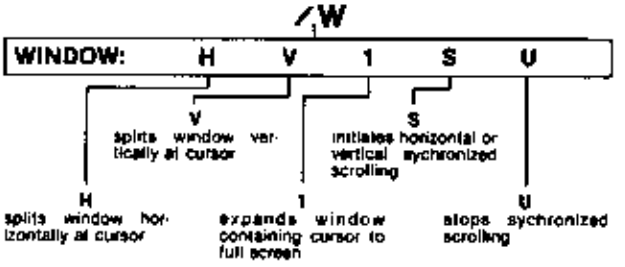
©1980 Software Arts Inc. 1.50

displays copyright notice and version number on prompt line, clears automatically with next keystroke

To see the copyright notice and the version number of your Visicalc program, type /V when the control panel is clear. You may use this command anytime, without disturbing the contents of a sheet you may have loaded. As soon as you press any key, the notice will disappear from the control panel.

Should you need to call or write with questions about VisiCalc, be sure to include the version number that appears when you give the version command. The photograph illustrates how the notice appears on the screen.

The WINDOW Command

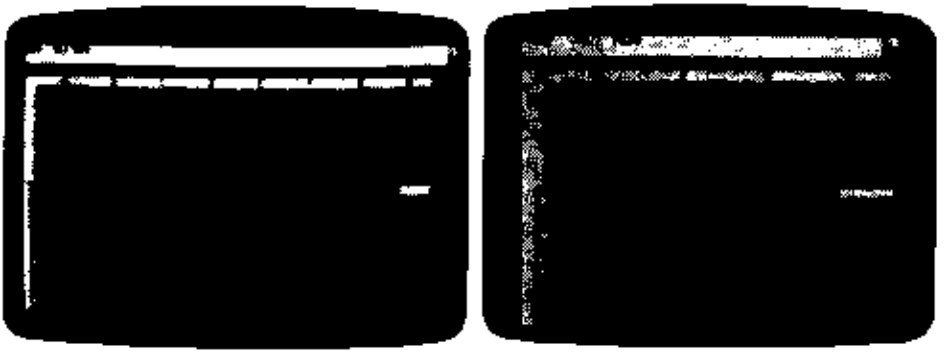


Often you will find yourself wishing to compare rows or columns which are too far apart on the sheet to be displayed in a single window on your computer screen. The window command allows you to split your screen so that you can view the sheet through two windows simultaneously. Each window may be independently scrolled around the entire sheet to let you see rows or columns which are widely separated on the sheet. You may also look at the same entry positions through separate windows with different global column widths and formats (see the GLOBAL Command) modifying the display in each one.

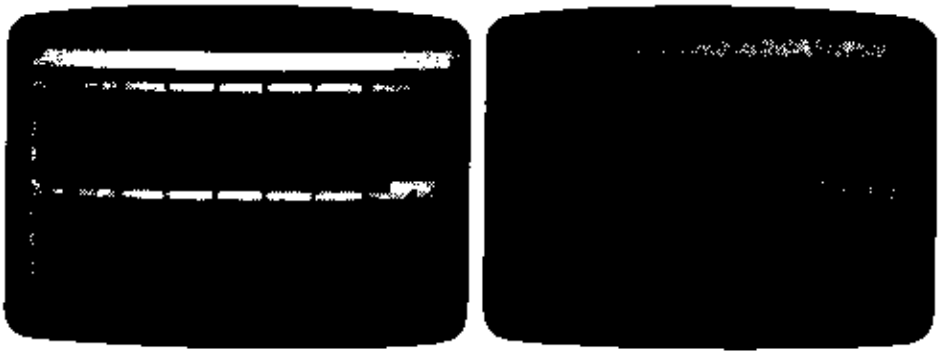
/WH (Window Horizontal) splits the window into two by placing a second column border (A,B,C,D. . .) between the row containing the highlighted entry position and the next row down. Each window may be moved individually to view the same or different parts of the sheet.

Example

1) Type /CY — This clears the VisiCalc sheet.
2) Type **ONE→TWO→THREE→FOUR→FIVE→SIX→SEVEN→EIGHT↩**
3) Type >A10↩
4) Type **JAN→FEB→MAR→APR→MAY→JUN→JULY→AUG↩** — Your screen should look like this:



3) Type /W prompt line: WINDOW: H V 1 S U
4) Type H Your screen will change to:



Each window will now view the sheet independently. The cursor can move around the sheet in the top window. Move it up and left until the ONE is highlighted in the top window. Press ; to move the cursor into the bottom window. Pressing ; will always make the cursor jump to the last position it had in the other window. (It will jump to the upper left entry position the first time it enters a window). All VisiCalc commands will work in both windows and you can see the effects on the sheet through either window. The two exceptions are the /GC (Global Column) and /GF (Global Format) commands, which are set in one window at a time (see the GLOBAL Command).

Remove the horizontal window by typing /W1. The window containing the cursor will then occupy the whole screen using the current format settings of that window. A horizontal window must be removed before a vertical window can be instated and vice versa. The size of each window is determined by the position of the cursor at the time the window command is used.

/WV (Window Vertical) splits the window by adding a second row border (1 2 3 4 5 . . .) after the column containing the cursor. When the screen has been split vertically, all the columns in the right window will be slightly narrower that those in the left window. This is to make room for the second row border. The vertical window behaves exactly like the horizontal window described above. Note that when you return to one window, the format settings of the window in which the cursor is sitting go into effect in the single window. The column width of the right window, which was automatically narrowed, will also be in effect if the cursor is in that window when /W1 is typed.

Example

1) Type /CY
2) Type
   JAN↵FEB↵
   MAR↵APR↵
3) Type >B1↵
4) Type
   100↵200↵
   300↵400↵
5) Type /WV

Your screen should resemble the photo below.

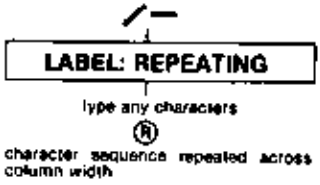
The vertical window can be particularly useful when you wish to keep a column of information visible in one window while you scroll the other window around to compare columns in different places on the sheet.

/W1 (One Window) displays the window containing the cursor in full screen position. All the global format settings in that window take effect in the one window.

/WS (Windows Synchronized Scrolling) synchronizes horizontal motion in horizontal windows or vertical motion in vertical windows so that moving the highlight in one window also moves the other.

/WU (window unsynchronized) turns off synchronized scrolling. The last three window command options (/W1, /WS, and /WU) may only be used after a /WH or /WV is in effect.

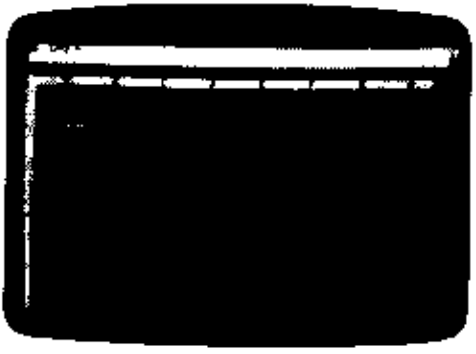
The REPEATING LABEL Command



Sometimes it is useful to draw lines or other borders across an entire column or across several columns. The repeating label command will repeat any sequence of characters you enter across the entire width of a column. If you change the column width (see the GLOBAL Command), the continuous sequence will be modified so that it still fills the column. When you replicate (see the REPLICATE Command) the entry containing the repeating label, you can form a line or other visual break across your sheet.

Example

1)	Position the highlight in the entry position to contain the repeating label.		
2)	Type /–	prompt line:	LABEL: REPEATING
3)	Type –	prompt line:	LABEL: REPEATING
		edit line:	–
4)	Press ®	VisiCalc will extend the sequence across the column. For example, with a general column width (9 characters) you see --------- in the entry position.	
5)	Use the replicate command to copy the continuous label as far across the sheet as needed.		



PART IV. INDEX

-A-

-B-

-C-

-D-

-E-

-J-

-K-

-L-

-M-

-N-

-O-

-P-

-Q-

-R-

-S-

-T-

-U-

-V-

-W-

-X, Y, Z-

-INDEX OF SPECIAL CHARACTERS-

$1.50

Software Arts Technical Note
SATN-18

PROGRAMMER'S GUIDE
to
DIF®
A Data Interchange Format

Software Arts, Inc.
4/81 V1.5

Limited License to Copy

No Warranty

This Guide is being published to enhance the usefulness of DIF, the data interchange format used by the VisiCalc® program and other programs. NEITHER SOFTWARE ARTS, INC. NOR PERSONAL SOFTWARE INC. MAKES ANY WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO THE QUALITY, ACCURACY OR FREEDOM FROM ERRORS OF THE DIF FORMAT OR OTHER CONTENTS OF THIS DOCUMENT, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR OF FITNESS FOR A PARTICULAR PURPOSE.

VisiCalc® a Trademark

The term "VisiCalc" is a trademark of Personal Software Inc. which designates a software product published by Personal Software Inc. under an exclusive license from Software Arts, Inc.

TABLE OF CONTENTS

1. INTRODUCTION

It is often desirable to process the same data by more than one program. For example, a data management system may be used to record sales values. These values are then to be used as the basis for projections using the VisiCalc program. Finally the projections may be plotted by a third program. How can you get data from one program to another without requiring the user to type the data in anew each time? Each of the programs processing the data may be written by a different person, and may even run on different machines.

In order to allow programs to "talk" to each other, we must agree upon a standard language. Software Arts, Inc., the creators of the VisiCalc program, have developed a data interchange format, DIF, that can be used as a common language for data. This is the format in which VisiCalc saves data with the */S#* commands.

We are writing this document in order to explain to programmers how they can read and write data files using this format. The more programs that support the format, the more useful it becomes. The casual user should not be concerned about the details. It is only important to be aware that the format exists and that if two programs support the format, then it is likely that data produced by one can be processed by the other.

If you read this document fully, you will learn all of the details of the standard. This is not a tutorial, so you may find it helpful to skim the more technical parts that follow, and concentrate on the next section, the beginning of the Data File Format section, and the sample programs.

The sample programs in this document are all coded in a general dialect of BASIC, except as noted. Files are opened with an OPEN statement, and read and written with INPUT# and PRINT# statements. To get these programs to run on your system, you may have to modify them.

2. WHAT DIF DOES AND DOES NOT DO

The basic goal of DIF is to allow the interchange of data among a wide variety of programs. The type of data addressed by DIF is data that is stored in tables—columns and rows. Examples of this type of data would be time series, such as the daily closing price of one or more stocks that are to be input to a regression analysis package, or the actual expense figures for a company that are to be used as the starting point for a forecast. DIF treats all data as a group of equal length *vectors*—that is, groups of related data, like time series, or columns in a relation. The word vector is used, rather than column, since the actual orientation of the data (a horizontal row or vertical column) does not necessarily correspond to how it is logically oriented. Likewise, the corresponding elements of the vectors are called *tuples* rather than rows. For example, in the data below, the Sales, Cost and Profit figures (across the rows) could be viewed as vectors, with each year (down the columns) corresponding to a tuple:

Year	1980	1981	1982	1983
Sales	100	110	121	133
Cost	80	88	97	106
Profit	20	22	24	27

The actual choice of which grouping of the data is considered to be the vectors, and which the tuples, is really up to the programmer or user. Some programs may just view the data as a rectangle of unrelated data, while others may require the user to be aware of the grouping. The VisiCalc program would be an example of the former, and a plotting package would be an example of the latter.

In DIF, data is stored by tuples. That is, it consists of successive values from each vector grouped together into tuples, which are then output (or input) in that order. In the data used for our example, if the vectors were across the rows (Sales would be one vector, Cost and Profit the other two), then the first tuple would consist of the three numbers 100, 80, and 20, in that order. The second tuple would be 110, 88, and 22, and so on.

When the VisiCalc program deals with data in DIF it gives you the option of storing or loading "by rows" (R or RETURN) or "by columns" (C). What the VisiCalc program means by "by rows" is that the vectors go across the rows, and the tuples go down the columns. For example, in our example data, saving Sales, Cost and Profit by rows would output first the tuple 100, 80, 20, and then the tuple 110, 88, 22, etc. "By columns" is just the opposite, with the vectors down the columns, and the tuples across the rows. For the same data, the first tuple by columns would be 100, 110, 121, 133, and then 80, 88, 97, 106, etc.

Not all of the programs that process the data stored in DIF will have identical requirements. For example, some programs will only be able to process a simple list of numbers while others will want to store attributes associated with multiple vectors of numbers. Thus, a goal in the design of DIF was that programs should be able to keep descriptive information about the data, but must not be *required* to generate it. At the same time, the program reading the data should be able to ignore all descriptive information that is not relevant to the actual processing of data.

The primary constraint on the format of data stored in DIF is simplicity. It should be very simple for users to write programs in a common language to read and write data files. Since BASIC is so pervasive and minimal, the needs of BASIC were used to determine the details of the format. It is necessary for other languages, such as Pascal or PL/I, to be able to process this data, too. Fortunately these languages allow the use of subroutine libraries. Thus, a standard set of subroutines to process the interchange format can be provided for the users of those languages, freeing them from many of the details of processing the data.

Nongoals were just as important as goals during the design of DIF. Specifically, there is no emphasis on a minimal space representation. This representation is meant to be modest and does not attempt to preserve the richness available in many database systems. The central idea is that we should be able to transport a table of values (numeric and/or string) from one program to

another. There is an additional mechanism to allow cooperating programs to exchange some information about the data, such as labelling.

Some of the more specific constraints are:

Predetermined data types

It is much simpler to write a program in BASIC if one knows ahead of time what the format of the data is, and in particular whether one is going to be reading a string or a number. Some BASICs are missing the VAL function that will convert from a string to a number, making it even more difficult. Therefore, DIF defines exactly which type of data is to be read at each point.

Lack of line input

Many BASICs do not have the ability to read a line of text without giving special meaning to some characters. For this reason strings containing special characters must be quoted.

Lack of parsing

Some BASICs will only input a whole line as a string. They do not use "," as a string value delimiter. Therefore, DIF always stores string values alone on a single line.

Input size

Many BASICs have a limited input buffer. 255 characters is a typical limit for the length of an input line. Therefore, DIF tries to keep most lines of information short.

Preallocation

In systems that permit dynamic allocation, it is often necessary to allocate the space before actually reading the data. Even when this is not required, knowing the total amount of data beforehand can be an important efficiency consideration. For this reason, DIF has a method for making this information available to a program reading the data.

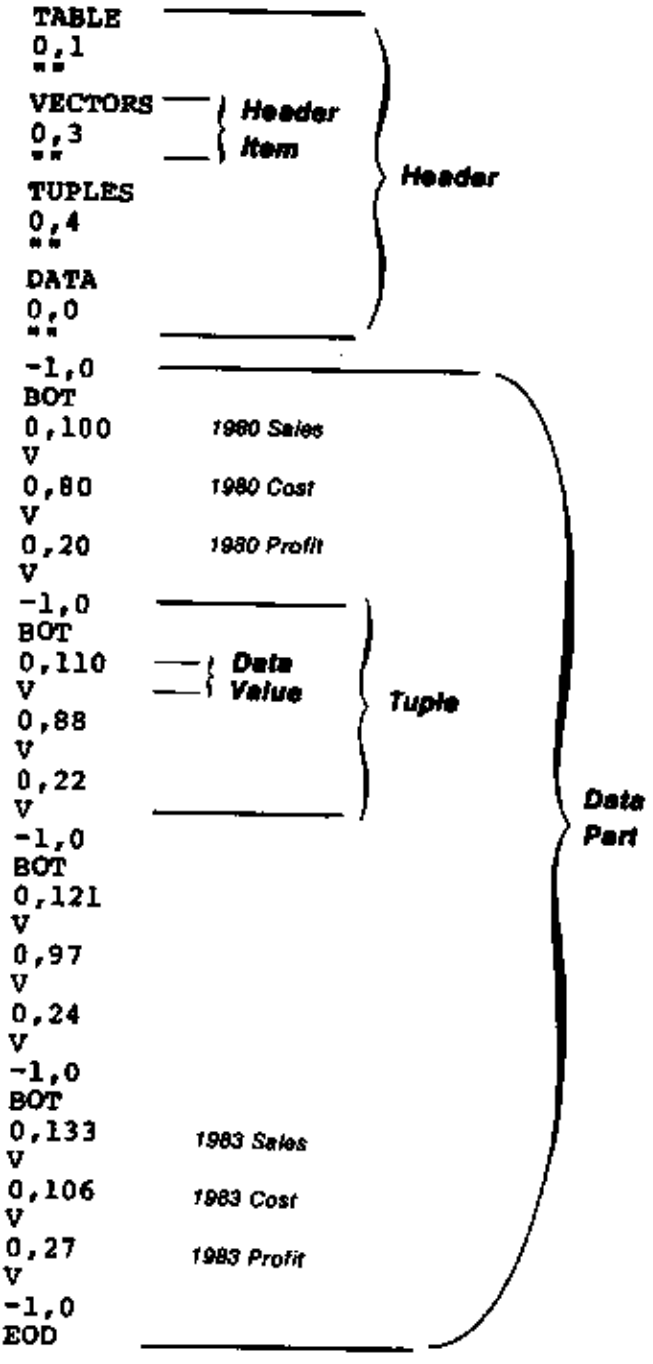
End of data

In some systems it is either difficult or impossible to detect the end of data in a file gracefully. Thus the program should know when it has read the last value. DIF has a special provision to signal when the last data element has been read.

3. THE DATA FILE FORMAT

A DIF file consists of two parts—the header and the data part. The header describes the data and the data part has the actual values. An example of a DIF

file is the following, which is from our sample data above. It has the vectors going across the rows, so there are three vectors, and four tuples. The various parts of the file are labelled, and will be described below:



THE HEADER

The header is organized into *header items*. Each header item contains a different piece of information about the data stored in the file. That data is sometimes numeric, and sometimes a string value.

STRUCTURE OF A HEADER ITEM

Each header item consists of four fields arranged as follows:

Topic
Vector number, Value
"String value"

The Topic

This is a keyword that identifies the header item. It must be a simple token readable as a string in BASIC without quotation marks. A word consisting of just letters with no spaces is best.

The Vector number

Several header items, such as a label, will apply to a specified vector. The Vector number specifies which vector this particular header item refers to. If the header item is not specific to a vector, such as a report title, this value should be 0.

The Value

This appears on the same line as the Vector number. It is used for header items that specify values, such as the number of vectors. It is zero if the value is not used by the header item. The value must be an integer.

The "String value"

This appears on a separate line after the Vector number and Value. It is used for header items that need string values rather than numeric values. The vector labels are an example. The string is always enclosed in quotes.

Thus the header item consists of three lines. The first line is the topic of the header item, the second line consists of two numbers and the third line has a string. The specific header items are described below.

Programs can ignore all header items until one with the topic DATA (described below) is found. The following program segment will skip the header items:

```
1000 INPUT#1,T$       :REM - Read the Topic name
1010 INPUT#1,S,N      :REM - Read the Vector #, Value
1020 INPUT#1,S$       :REM - Read the String value
1030 IF T$<>"DATA" THEN 1000 :REM - Check for
                             :REM - DATA header item
```

THE HEADER ITEMS

The standard header items are shown below with a description. The only required header items are TABLE and DATA, which must be the first and last header items, respectively.

TABLE
0,version
"title"

This is the first entry in the file. While it is not strictly required, it is important to allow for changes in future versions and it allows programs to verify that the file is a TABLE of data. The version number must be 1. Some programs may not accept the file without the TABLE header item.

VECTORS
0,count
" "

This tells how many data vectors are present. Some programs will require this header item to be present. If this header item is absent, the input program can calculate this value by counting the number of Data Values in each tuple (see below). N.B.: This header item must appear before header items that reference vector numbers, such as the LABEL header item.

TUPLES
0,count
" "

Specifies the length of each vector. (All vectors must be the same length.) Some programs will require this header item. If this header item is absent, the input program can calculate this value by counting the number of tuples before an end of data (EOD) Special Data Value (see below).

LABEL
vector#,line#
"label"

Provides a label for the specified vector. This is optional. The line# allows for labels spanning multiple lines, but can be ignored by systems allowing only single line labels. The values 0 and 1 should be equivalent for line#.

COMMENT
vector#,line#
"label"

This is similar to the LABEL header item for systems that allow an expanded description in addition to labels.

SIZE
vector#,#bytes
" "

This is used by programs, such as data base systems, that allocate fixed size fields for each value. Such programs, though, should be able to read files that do not contain SIZE information, since other programs may not be able to generate information of this type.

DATA
0,0
" "

This says that data follows. The data is organized by tuples, with one value from each vector in a given tuple.

Subsystems may define their own header items to meet their needs. Header items that will tend to be common should be standardized.

such as the LABEL for a vector. The DIF Clearinghouse will serve as a repository for standard header items (see the address for DIF correspondence in the section Clearinghouse, below).

THE DATA PART

The data part consists of tuples, i.e. one value for each vector, in vector order. The tuples are made up of groups of two numeric values and one string value called *Data Values*. Each Data Value is used to represent the value of one element of data in the file.

In addition to the Data Values used to represent the actual data in the file, there are two types of Special Data Values used to provide information about the organization of the data. One Special Data Value is used to show where each tuple starts, and the other Special Data Value is used to indicate the end of all of the data in the file.

Data Values are all in the following format:

Type Indicator, Number Value
String Value

The first two fields are numeric values on a single line, the last is a string on a line by itself. These fields are:

The Type Indicator field

The Type Indicator is an integer that is used to indicate the way in which to interpret the rest of the fields in a Data Value. The currently assigned values for the Type Indicator are:

-1 Indicates that this Data Value is a Special Data Value, either a beginning of tuple indicator or an end of data indicator. See below for a discussion of the Special Data Values.

0 The data is numeric. The value of the Data Value is stored in the Number Value field, possibly modified by the String Value (see the descriptions of the Number Value and String Value fields below).

1 The data is a string. The value of the Data Value is stored in the String Value field.

2 This is an application specific value. The meaning is determined by the cooperating programs that are expected to use the data. For example, it might be an expression in the host language. For simple applications these values can be treated as strings.

The Number Value field

This is used when the Type Indicator is 0 to represent the value. The value must be a decimal (base 10) number. It may optionally be preceded by a sign (+ or –), have a decimal point, and immediately be followed by the letter E and an optionally signed power of ten expo-

nent. The number may be preceded or followed by one or more blanks. Note that this is the only place in DIF where a non-integer value is allowed. Some programs that read data in DIF may only accept integer values (e.g., programs written in some BASICs or some systems programming languages).

The String Value field

The interpretation of this field depends upon the Type Indicator.

For normal Type Indicator 0 (numeric) data, the String Value should be the letter V (for value). If it is not V, then it is a Value Indicator, used to override the value. A subsystem may choose its own Value Indicators for named values, though they should be registered with the DIF Clearinghouse. The following Value Indicators are used by the VisiCalc program:

V

This is the normal case for numbers.

NA

This is a value marked explicitly as Not Available. The Number Value is set to 0.

ERROR

This is a value that represents the result of an invalid calculation, such as division by 0. The Number Value is set to 0.

It should always be possible to ignore the String Value for numeric data and just use the Number Value given. Another simple approach is to treat all values with a Value Indicator other than "V" as missing. Note that quotes are not permitted around the Value Indicator (for the sake of some BASICs).

For the Type Indicator of 1 (string data), this field is used for the string value itself. The quotes are optional if the field consists of just letters and does not contain any spaces. However, if a starting quote is given, a terminating quote must also be given.

Each tuple begins with a Special Data Value whose Type Indicator is -1, Number Value is 0, and whose String Value is BOT (for Beginning Of Tuple). This Special Data Value can be used by programs to determine how many vectors are in the file in the absence of a VECTORS header item (by counting the number of Data Values between BOT Special Data Values), or for a program to verify its position in a file.

At the end of the last tuple is a Special Data Value with a Type Indicator of -1, a Number Value of 0, and a String Value of EOD (for End Of Data). This will allow programs to determine the number of tuples in the absence of a TUPLES header item (by counting the number of tuples before an EOD Special Data Value), and to gracefully detect the end of the file.

4. SAMPLE PROGRAMS

Here are two sample programs. The first program creates a DIF file. The second program can read a DIF file and list its contents. They should be helpful in understanding how to manipulate DIF files. They are written as main programs with subroutines, so you can pick up code from them to be used in other programs. Both programs are written in a general BASIC, as described above.

CREATING A DIF FILE

```
100  REM - This program creates a  DIF  file.
110  REM - It prompts for the file name, number of vectors and
120  REM - tuples, and then for the values themselves.  Data
130  REM - may be either numeric (type 0) or string (type 1).
140  REM
1000 PRINT "FILE NAME";                  :REM - Get name of file
1010 INPUT F$
1020 OPEN 1,F$                           :REM - Open for write
1030 PRINT "NUMBER OF VECTORS";          :REM - Get number of vectors
1040 INPUT NV                            :REM - into variable NV
1050 PRINT "NUMBER OF TUPLES";           :REM - and number of tuples
1060 INPUT NT                            :REM - into variable NT
1070 GOSUB 3000                          :REM - Write out DIF header
1080 FOR I = 1 TO NT                     :REM - Get data and output it
1090   T = -1: V = 0: S$ = "BOT"         :REM - Output beginning of tuple
1100   GOSUB 4000
1110   FOR J = 1 TO NV                   :REM - Get each Data Value
1120     PRINT "DATA TYPE FOR VECTOR #";J;", TUPLE #";I;
1130     INPUT T
1140     V = 0: S$ = "V"                 :REM - Init values
1150     PRINT "DATA VALUE FOR VECTOR #";J;", TUPLE #";I;
1160     IF T=0 THEN INPUT V
1170     IF T=1 THEN INPUT S$
1180     GOSUB 4000                      :REM - Output the Data Value
1190     NEXT J
1200   NEXT I
1210 T = -1: V = 0: S$ = "EOD"           :REM - Output end of data
1220 GOSUB 4000
1230 CLOSE 1
1240 PRINT "FINISHED CREATING DIF FILE ";F$
1250 STOP
3000                          :REM - Routine to write out DIF header
3010 PRINT#1,"TABLE": PRINT#1,"0,1": GOSUB 3500
3020 PRINT#1,"TUPLES": PRINT#1,"0,";NT: GOSUB 3500
3030 PRINT#1,"VECTORS": PRINT#1,"0,";NV: GOSUB 3500
3040 PRINT#1,"DATA": PRINT#1,"0,0": GOSUB 3500
3050 RETURN
3500                          :REM - Routine to write "" (null string)
3510 PRINT#1,CHR$(34);CHR$(34)       :REM - See Appendix on quoted
3520 RETURN                          :REM - strings in BASIC, below
4000                          :REM - Routine to write out Data Value
4010 PRINT#1,T;",";V
4020 PRINT#1,S$
4030 RETURN
4040 END
```

Note that if the string values being saved have spaces or special characters, the code at line 4020 should be changed to check for those cases, and add leading and trailing quotes. See the discussion about Quoted Strings in BASIC in the Appendix.

LISTING A DIF FILE

```
100  REM - This program reads a DIF file
110  REM - and lists its contents.  The program prompts for
120  REM - the name of the file to be listed.
500  DIM T(100)                          :REM - Maximum of 100 vectors
510  DIM V(100)                          :REM - T, V, and V$ hold the
520  DIM V$(100)                         :REM - Type Indicator, Number
530                                      :REM - Value and String Value
540                                      :REM - of each element in a tuple
550                                      :REM -
1000 GOSUB 5000                          :REM - Call initialization code
1010 GOSUB 6000                          :REM - Read header
1020 FOR I = 1 TO NT                     :REM - Read all of the tuples
1030   PRINT "VALUES FOR TUPLE #";I
1040   GOSUB 7000                        :REM - Get a tuple
1050   FOR J =1 TO NV                    :REM - Output each element
1060     IF T(J)=0 THEN PRINT V(J)       :REM - Output numeric value
1070     IF T(J)=1 THEN PRINT V$(J)      :REM - Output string value
1080     NEXT J
1090   NEXT I
1100 CLOSE 2
1110 PRINT "FINISHED LISTING FILE ";F$
1120 STOP
5000                          :REM - Initialization code
5010 PRINT "FILE NAME";                  :REM - Get name of file to read
5020 INPUT F$
5030 OPEN 2,F$                           :REM - Open file for read
5040 NV = 0                              :REM - Init counts of vectors
5050 NT = 0                              :REM - and tuples
5060 RETURN
6000                          :REM - Read header, and set NV and NT
6010 INPUT#2,T$                          :REM - Get Topic name
6020 INPUT#2,S,N                         :REM - Get Vector number, Value
6030 INPUT#2,S$                          :REM - Get "String value"
6040 IF T$="VECTORS" THEN 6500           :REM - Check for known header
6050 IF T$="TUPLES" THEN 6600            :REM - items
6060 IF T$="DATA" THEN RETURN            :REM - DATA ends header
6070 GOTO 6010                           :REM - Ignore unknown ones
6500 NV = N                              :REM - Value is number of vectors
6510 PRINT "THE FILE HAS ";NV;" VECTORS."
6520 IF NV<=100 THEN 6010                :REM - If not too many, continue
6530 PRINT "TOO MANY VECTORS.  THIS PROGRAM ONLY HANDLES 100."
6530 CLOSE 2
6540 STOP
6600 NT = N                              :REM - Value is number of tuples
6610 PRINT "THE FILE HAS ";NT;" TUPLES."
6620 GOTO 6010                           :REM - Get next header item
7000                          :REM -  Get all vector elements in a tuple
7010 GOSUB 8000                          :REM - Get next Data Value
7020 IF T1<>-1 THEN 9000                 :REM - Must be BOT or else error
7030 IF S$<>"BOT" THEN 9000
7040 FOR K = 1 TO NV                     :REM - Get each Data Value
7050   GOSUB 8000
7060   IF T1=-1 THEN 9000
7070   V(K) = V1                         :REM - Save Values and Type
7080   V$(K) = S$                        :REM - Indicator
7090   T(K) = T1
7100   NEXT K
7110 RETURN
8000                          :REM - Get Next Data Value
8010 INPUT#2,T1,V1                       :REM - Get Type Indicator,
8020 INPUT#2,S$                          :REM - Numeric Value and String
8030 RETURN                              :REM - Value
```

```
9000 PRINT "ERROR IN FILE FORMAT."
9010 CLOSE 2
9020 STOP
9030 END
```

Please note that while the above program can read many DIF files correctly, it depends upon the TUPLES and VECTORS header items to determine the organization of the file. A more general program could be written that, in the absence of these header items, deduced their values from the placement of BOT and EOD Special Data Values. While most programs that deal with DIF should be able to produce TUPLES and VECTORS header items (the VisiCalc program, for example, does), some may not (such as a program that records data incrementally, and doesn't know how many data points it will encounter until it is finished).

5. APPENDICES

QUOTED STRINGS IN BASIC

Writing the quoted strings is not always convenient in BASIC. In some implementations, quotes may be included in a string by doubling them. For example:

```
PRINT#1,"TABLE"
PRINT#1,0,1
PRINT#1,""Stock Prices for ABC Computer Co.""
```

In other implementations the CHR$ function must be used:

```
PRINT#1,"TABLE"
PRINT#1,0,1
PRINT#1,CHR$(34);"Stock Prices for ABC Computer Co.";CHR$(34)
```

Apple Integer BASIC presents special problems. It seems that it is necessary to POKE an assembly language routine into memory to output a quote. The following sequence will setup such a program at location $300 (hex):

```
100 POKE 768,169:POKE 769,162          :REM - LDA #'"'+$80
110 POKE 770,108:POKE 771,54:POKE 772,0 :REM - JMP (CSWL)
```

And to use this code:

```
120 PRINT "TABLE"
130 PRINT 1,0
140 CALL 768
150 PRINT "Stock Prices for ABC Computer Co.";
160 CALL 768
170 PRINT
```

Apple Integer BASIC also requires that the user remove the quotes from the input string with:

```
300 IF LEN(S$) > 2 THEN
        IF (ASC(S$(1,1)) MOD 128) = 34 THEN
            S$ = S$(2,LEN(S$)-1)
```

This assumes that there is also a trailing quote. Note that in order to make the quoted string itself acceptable to most BASICs it must not contain a quote.

CHARACTER SETS

The character set is assumed to be that of the host machine. Thus, if one is transferring a file from a machine using ASCII to one using EBCDIC, the appropriate conversions must be made. In addition, some machines may require that the quote be changed to an apostrophe. These changes should be transparent to most users. In order to assure compatability, strings should not contain nonprinting characters, other than the end of line sequence (RETURN, CR/LF, NEWLINE or whatever).

The ASCII character set defines 95 printable characters. The user should be aware that some systems do not make it easy to use the full set. In particular, keywords (including topic names and number types) must be in upper case. Some systems only support a limited set of characters, often 64 printable characters or less. When transporting a file to such a system the upper and lower case characters would be mapped together to one case. Other special characters may be mapped into common characters. If these transformations affect the integrity of the data, it should be specified in the documentation associated with the data.

6. CLEARINGHOUSE

In order to coordinate information about DIF and the programs that make use of it, Software Arts, Inc. is setting up a clearinghouse for such information.

We would appreciate it if the authors of programs that support DIF would send a one page description of the program to the clearinghouse. This description should include a short write-up of what the program does, on which computers it runs, how it relates to DIF, and how it may be obtained.

Users who would like a copy of the information that we receive should send $6.00 (to cover the costs of running the clearinghouse and providing the information) and their name, address and zip code, with a note specifically requesting a copy of the list of programs that support DIF.

All correspondence relating to DIF should be sent to the following address:

DIF Clearinghouse
P.O. Box 527
Cambridge, MA 02139

VISICORP®
PERSONAL SOFTWARE™

2895 Zanker Road
San Jose, California 95134
Telephone: (408) 946-9000