PLAN80

A powerful tool for planning and analysis

Business Planning Systems

CHAPTER 1

INTRODUCTION

PURPOSE OF PLAN80

PLAN80 is a powerful tool for significantly improving the quality and timeliness of plans, forecasts, budgets and analyses through the automated preparation of polished reports on a computer screen or printed page.

SCOPE OF THIS DOCUMENT

This manual will show you how to use PLAN80 for a broad range of planning and analytical problems. PLAN80 lets you in concise, easy-to-understand terms

* describe a problem's framework in terms of rows and columns,
* input data values,
* specify calculations,
* display and/or print results, and
* communicate results between PLAN80 applications.

HOW TO PROCEED

Read Chapters 1 and 2 of this manual. Then skim Chapters 3 through 8. You'll have a good overview, and find it easy to use the manual subsequently as a detailed reference.

Read Appendix B before operating PLAN80.

Learn to operate PLAN80 from Appendix C and use the examples in Appendix A as a pattern for your own applications.

PLAN80 should increasingly help you fulfill your potential as an imaginative problem solver by eliminating the drudgery of manipulating numbers and presenting results.

ORGANIZATION OF THIS MANUAL

Chapter 2 explains PLAN80'S capabilities and uses and provides an overview of its features.

The remaining chapters present for each PLAN80 statement
* format and content of the statement,
* one or more examples,
* detailed discussion of content, and
* discussion of statement usage.

Chapter 3 covers the framework of an application:
* TITLES
* COLUMNS
* ROWS

Chapter 4 explains how to establish data values.
* DATA
* INITIALIZE

Chapter 5 shows you how to calculate results.
* RULES

Chapter 6 discusses statements which print or display results or allow you to use the interactive features of PLAN80.
* PRINT
* DISPLAY
* OPTIONS

Chapter 7 explains statements which permit communications between PLAN80 applications.
* PUT
* GET

Chapter 8 covers additional PLAN80 statements.
* INTERACTIVE
* INCLUDE
* REPEAT
* FOR

Appendix A contains examples to help you understand PLAN80 and to use as a pattern for your own applications.

Appendix B discusses what you must do before operating PLAN80.

Appendix C tells you how to operate PLAN80 on your computer.

Appendix D contains a description of error codes produced when PLAN80 detects a problem in your file of input statements.

Appendix E discusses memory and diskspace considerations.

CHAPTER 2

A PLAN80 OVERVIEW

WHAT IS PLAN80?

PLAN80 is a modeling system that helps you

* plan,
* forecast,
* project,
* estimate,
* analyze,
* control, and
* understand

numbers representing

* sales,
* profits,
* costs,
* taxes,
* cash,
* marketing plans,
* cost center expense,
* R&D projects,
* market share,
* growth rates,
* return on investment,
* capital projects,
* real estate deals, and
* discounted cash flow.

Whatever your application

* sales forecasting and analysis
* profit and loss analysis
* balance sheet pro_ections
* long range planning
* cash flow forecasting and analysis
* cost center budgeting and analysis,
* project budgeting and control
* headcount control and analysis
* salary and wage analysis
* purchase price analysis
* cost and variance analysis
* market share planning and analysis
* capital project evaluation

you will probably want the results presented as a tabular report. And PLAN80 automates the preparation of tabular reports.

WHY A COMPUTER?

Without a computer, planning and analysis invariably mean working with hundreds, maybe thousands, of numbers. Assumptions and calculations must be checked and results summarized again and again. A difficult problem may take hours or days.

If only someone else could do the number crunching and copying. You could spend your time more productively on other tasks. Well that's what PLAN80 does for you. It relieves the drudgery of calculating, recalculating, checking and producing high quality typed reports.

HOW DOES PLAN80 WORK?

Every tabular report has certain common elements. Imagine that you are instructing an assistant to prepare a report. You would provide information about its TITLES, its framework in terms of COLUMNS and ROWS, the starting DATA values, and the RULES used to compute totals and other values. Also you would specify items such as column width, line spacing, the number of decimal positions, etc., to make the report visually effective.

You use the PLAN80 language to instruct your computer with this same information. Since your computer is not as bright as a human assistant you must be very precise. But your electronic helper is extremely fast and unfailingly accurate.

A PLAN80 EXAMPLE

Let's look at a PLAN80 example. The following report was prepared from the first example included on your PLAN80 distribution disk. It projects net income as a function of sales, costs and taxes and calculates margins for gross profit, operating profit and net income.

THE OUTPUT

```
                    PLAN80 EXAMPLE #1
               Five Year Profitability Model

                   1981     1982     1983     1984     1985


ASSUMPTIONS
  Units             100      118      142      170      204
  Price            1.32     1.49     1.64     1.80     1.98
  Unit Cost       0.818    0.919    1.029    1.153    1.291
  Tax Rate        0.460    0.460    0.460    0.460    0.460
DOLLARS (THOUSANDS)
  Sales             132      176      232      306      404
  Cost of Sales      82      108      146      196      263
  Overhead           19       19       21       22       24
                  -----    -----    -----    -----    -----
  Profit Before Tax  31       48       66       88      117
  Taxes              14       22       30       41       54
                  -----    -----    -----    -----    -----
  Net Income         17       26       36       48       63
                  =====    =====    =====    =====    =====
MARGINS (PERCENT)
  Gross Profit     38.0     38.3     37.2     36.1     34.9
  Operating Profit 23.6     27.5     28.4     28.8     29.0
  Net Income       12.8     14.9     15.3     15.6     15.6
```

THE INPUT

The above report resulted from the following PLAN80 statements:

```
:TITLES
  1 "PLAN80 EXAMPLE #1"
  2 "Five Year Profitability Model"
:COLUMNS
  Y1981   "1981"
  Y1982   "1982"
  Y1983   "1983"
  Y1984   "1984"
  Y1985   "1985"
:ROWS
  UNITS             "ASSUMPTIONS"
                    "  Units"
  PRICE   (2)       "  Price"
  UCOST   (3)       "  Unit Cost"
  TAXRT   (3)       "  Tax Rate"
  SALES             "DOLLARS (THOUSANDS)"
                    "  Sales"
  COST              "  Cost of Sales"
  OVERHEAD          "  Overhead"
  PBT     (-)       "  Profit Before Tax"
  TAXES             "  Taxes"
  NET     (-=)      "  Net Income"
  GPM     (1)       "MARGINS (PERCENT)"
                    "  Gross Profit"
  OPM     (1)       "  Operating Profit"
  NPM     (1)       "  Net Income"
:DATA
  UNITS     =  100   118   (*1.20)
  PRICE     = 1.32  1.49   (*1.10)
  UCOST     = .818  .919   (*1.12)
  TAXRT     =  .46    +
  OVERHEAD  =   19    19   (*1.08)
:INTERACTIVE
:RULES
  SALES     = UNITS * PRICE
  COST      = UNITS * UCOST
  PBT       = SALES - COST - OVERHEAD
  TAXES     = TAXRT * PBT
  NET       = PBT - TAXES
  GPM       = 100 * (SALES - COST) / SALES
  OPM       = 100 * PBT / SALES
  NPM       = 100 * NET / SALES
:OPTIONS
  ROWWID(19)
:DISPLAY
```

COMMENTARY ON EXAMPLE 1

Note that the PLAN80 statements are divided into sections (TITLES, COLUMNS, ROWS, DATA and RULES) similar to those used in the general description of tabular reports.

The TITLES section introduces the two title lines at the top of the report.

Columns and rows are defined in COLUMNS and ROWS sections. Each definition begins with a name, optionally followed by numbers or characters within parentheses. The numbers control the decimal positions shown, and the other characters control under any overlining and other printing options. Printed descriptions are contained in quotes.

In DATA statements values may be assigned to rows on a column by column basis or to columns on a row by row basis. Furthermore, the values may be individually specified or calculated as part of a series. For example, units are 100 and 118 for the first two periods and increase at a 20% rate thereafter, and the tax rate is 46% for all periods.

The INTERACTIVE statement indicates where PLAN80 should begin to relaculate after values are changed in the DISPLAY mode.

The RULES statements indicate calculations: sales as units times price, cost as unit cost times sales and profit before tax as the result of sales less cost and overhead. Taxes are a result of the tax rate times profit before tax, and net income equals profit before tax less taxes. The margins are calculated by dividing various quantities by sales, and are multiplied by 100 to convert the results to percentages.

The OPTIONS section lets you specify characteristics of a printed report. In this instance the row description field is to be 19 characters wide.

The DISPLAY statement allows you to view results on your computer screen or to print the report as was done for the example. It also lets you input new data values interactively and repeat all statements occurring after the INTERACTIVE statement.

The PRINT statement causes printing of reports without further operator intervention. The OPTIONS feature may be used to change printing options.

The PUT statement allows you to save portions of the values within a PLAN80 application on disk for subsequent use by a GET statement. This feature permits PLAN80 to consolidate financial statements or support interrelated planning models which feed into a summary model.

OTHER FEATURES

The above PLAN80 elements enable you to create powerful applications for planning and analysis. PLAN80 has still other features which will enhance its value to you.

- Results may be graphed as well as displayed in numeric form on your screen.

- Data may be changed interactively. This is particularly useful in examining the sensitivity of results to alternative assumptions about data values.

- You may INCLUDE external files that contain PLAN80 statements. This allows one set of column or row definitions to be used by many applications and it allows one application to handle an unlimited number of sets of data, as would be required in building cost center budgets for a large number of cost centers. This feature may also be used in examining the effects of many different groups of data or calculation assumptions.

- You may REPEAT external files containing PLAN80 statements. This feature is similar to the INCLUDE feature, but it allows you to repeat a group of statements any number of times.

- Using the PUT and GET statements it is possible to construct models of any size.

ABOUT PLAN80 SYNTAX

Each PLAN80 statement must begin on a new line, but the statement may begin in any column.

Statements may be continued on any number of lines, but elements, such as names, numbers, descriptive data in quotes, etc., may not be divided by blanks or line breaks.

Any number of blanks may occur between the elements of a PLAN80 statement.

USE OF LOWER CASE CHARACTERS

You may freely intermix upper and lower case letters throughout PLAN80. Thus you may use the name

```
    "SALES" in a row definition,
    "Sales" in a DATA statement, and
    "sales" in a  RULE.
```

Control statements such as :COLUMNS, :ROWS, :DATA, :RULES, etc., may also be written in lower case. Also keyboard response may be in upper or lower case.

COMMENTS

A PLAN80 statement listing is usually self documenting. However, comments are sometimes desirable. If a line begins with a colon followed by a blank then the line is treated as a comment. If a colon occurs after a blank and not within quote marks then all characters to its right are treated as a comment.

```
  : THIS LINE IS A COMMENT
  INCOME = SALES - EXPENSE     :THIS IS A COMMENT
```

STATEMENT LISTING APPEARANCE

PLAN80 has few format restrictions. You should use indentation, spacing and multiple lines to achieve a visually pleasing listing of your PLAN80 statements. It will then be easier to review your assumptions and calculations with others having no knowledge of PLAN80. And months after you have last looked at an application you will understand it more quickly in order to update it for current needs.

CHAPTER 3

DEFINING THE FRAMEWORK

A "TITLES" SECTION

Title statements must occur within a titles section headed

```
  :TITLES
```

Each title statement contains

- the relative line number,
- whether the title is to be RIGHT or LEFT justified, if not centered, and
- the characters comprising the title line, enclosed in quotes.

Each title statement must begin on a separate line and may not extend to a second line.

AN EXAMPLE

The following statements define a four line title with the first line printed on the left and the next three centered.

```
  :TITLES
    1 LEFT "3/31/85"
    2 "Janus Masks, Inc."
    3 "SALES ANALYSIS (1979-1984)"
    4 "(Thousands of Dollars)"
```

RELATIVE LINE NUMBER

A relative line number is used to control the placement of title lines in a title that may be up to nine lines long.

Titles may be double spaced by using every other line number as in 1,3 and 5. Lines 2 and 4 in this case would be printed as blank lines. If the last line noted is 5, only five lines will be printed.

In applications using only a single title line the line number is optional. Whenever a line number between 2 and 9 is not provided, PLAN80 assumes the new title applies to the first title line.

LEFT OR RIGHT JUSTIFICATION

The data on the first line of the example above would be justified left and the other three lines would be centered between the first and last columns of each printed page. Any line could be moved to the far right by placing the word "RIGHT" between the line number and the title data.

USE OF QUOTE MARKS

Title data must occur between double quotes. Any printable character may be entered between the quotes, except, of course, a double quote. Each line is limited to 60 characters.

REDEFINING TITLE LINES

In complex applications, as in a sensitivity or "what if?" analysis, it is possible to redefine one or more title lines used previously. For example, the first several lines of an application may be standard ones (company name, project name, etc.) and the last could be used to introduce a description of the case or situation being analyzed. It could then be redefined for each case.

In replacing a previous line there is no need to "erase" the original line. The new one will overlay the first completely, even if the second is shorter than the first or is placed at a different position on the page (e.g., justified right instead of centered).

ELIMINATING TITLE LINES

To eliminate all titles defined previously you may place the word "RESET" after the titles statement as in

```
  :TITLES RESET
```

or to eliminate a single line without adding any characters to be printed you may replace a line's contents with "nothing" as in the following example

```
  :TITLES
    3 ""
```

This example would cause line 3 to be printed as a blank line.

A "COLUMNS" SECTION

Column statements must occur within a columns section headed

```
:COLUMNS
```

Each column statement must contain

* the internal name of the column, which is used in data, calculation and certain other control statements.

In addition any column statement may contain

* an indicator "P" if the column is to begin on a new page when results are being printed,
* the minimum number of decimal positions to be displayed for all values within a column,
* an indicator "T" if the column is to contain only calculated values,
* an indicator "X" if the column is not to be printed, and
* column headings of up to three lines with each line containing up to eight characters within double quote marks.

Each column statement must begin on a new line, and may continue on subsequent lines.

AN EXAMPLE

The following lines define four columns:

```
:COLUMNS
  A1981        "ACTUAL"  "1981"
  P1982 (2)    "  PLAN"  "1982"
  P1983 (P3)   "  PLAN"  "1983"
  TOTAL (3)
```

COLUMN NAMES

The four columns above have the names A1981, P1982, P1983 and TOTAL. Names must begin with an alphabetic character ("A" through "Z") and may be as long as eight characters. The second and subsequent character may be numeric ("0" through "9") as well as alphabetic. You may not use any other characters. You should use meaningful names, such as JAN, QTR2, ALT3, etc.

PAGE BREAKS

The above columns would be printed on two separate pages because of the "P" indicator in the third line. The "P" causes the column named COLC to begin on a new page. (Note: the printer will automatically provide a new page for the first column.)

DECIMAL POINT SPECIFICATION

The first column has no decimal point specification and the others specify that two or three positions be printed to the right of the decimal point. The specification affects only the way that values are printed, never the values themselves.

SPECIFICATION PRECEDENCE

If there is a decimal point specification for both a row and a column then the larger of the two specified values will be used. For example, if ROWA has a specification of three positions and COLX has a specification for none, one or two positions, then the value printed at the intersection of ROWA and COLX will show three positions to the right of the decimal point.

SUMMARY OF PRINT SPECIFICATIONS FOR COLUMNS

Page and decimal point information must be specified between parentheses "(" and ")" after the column name and before the double quote which introduces descriptive headings. Between the parentheses may be any of the following:

"P" - indicates that a new page should occur before the column is printed.

"n" - a digit between "1" and "6" to specify the minimum number of positions to be printed to the right of a decimal point. When this parameter has been specified for both rows and columns the larger prevails. No decimal point will be printed for a value whose row and column specifications have no decimal point indicator.

"T" - indicates that the column is generally a calculated or "total" column and all values may be set to zero using the "z" (zero) command in the display mode.

"X" - sometimes it is convenient to set aside temporary columns to hold intermediate calculation results. An "X" causes PLAN80 to ignore the column during printing. The column will, however, be displayed on the screen for possible confirmation of intermediate results.

EFFECT OF UNUSED HEADING LINES

The first three lines of the example above use two of the three available column headings, and the third will be printed in each case as a blank line as though the specification had been:

```
:COLUMNS
  COLA          "ACTUAL"  "1981"  "    "
  COLB  (2)     "  PLAN"  "1982"  "    "
  COLC  (P3)    "  PLAN"  "1983"  "    "
```

The fourth line of the original example is a special case in which no descriptive information is provided. The letters "TOTAL" in this case are used as though the original specification had been:

```
  TOTAL       " TOTAL"  "    "  "    "
```

Anytime that descriptive information is not explicitly provided the column name is used as a one line description.

CONTROLLING COLUMN HEADING APPEARANCE

The descriptive information between double quotes may contain any character except a double quote, including leading and trailing blanks. When reports are prepared the description of each line is right justified as shown in the following example:

```
COLN    "ACTUAL"  "1ST"  "QUARTER"
```

would print as

```
 ACTUAL
    1ST
QUARTER
```

It is possible to form intermediate headings, covering groups of columns, as in the following example:

```
M1     "     ---"  "JAN"
M2     "------ 1"  "FEB"
M3     "ST QTR -"  "MAR"
Q1     "--------"  "TOTAL"
```

which would print as

```
--------- 1ST QTR ---------
JAN     FEB     MAR   TOTAL
```

WHEN NO DESCRIPTION IS PROVIDED

The column name is used as a description if no description is provided. In other words, the following are equivalent.

```
COL99
COL99    "COL99"
```

PLACEMENT OF A COLUMNS SECTION

Normally a COLUMNS section will occur only once in a PLAN80 run. At least one column must be entered before any rows may be defined or data or calculations entered. Only TITLES and OPTIONS (for print) sections may precede the first COLUMNS section.

If a second COLUMNS section is encountered then PLAN80 assumes you are beginning a new application within the same physical run. This is permissible and is an effective way to handle complex runs which require no operator intervention. All prior defined columns, data and previously computed values will be forgotten. Any TITLES and OPTIONS will survive the introduction of a new COLUMNS section, however.

A "ROWS" SECTION

Row statements must occur within a rows section headed

```
:ROWS
```

Each row statement must contain
- the internal name of the row, which is used in data, calculation and in certain other control statements.

In addition any row statement may contain
- an indicator "P" if the row is to begin on a new page when results are printed,
- an indicator "S" for each blank line, up to three, that is to occur before the row is printed,
- an indicator "-" or "=" if the row should have an over or underscore to highlight totals,
- the minimum number of decimal positions to be displayed for all values within the row,
- an indicator "T" if the row is to contain only calculated values,
- an indicator "X" if the line is not to be printed, and
- row heading information which can occur on any number of lines.

Each row statement must begin on a new line, but may continue on subsequent lines.

AN EXAMPLE

The following lines define nine rows:

```
  :ROWS
    ROWA     (P1)      "EASTERN REGION"
                       " NEW YORK"
                       "  ALBANY"
    ROWB     (1)       "  BUFFALO"
    ROWC     (1)       "  CHAMPLAIN"
    TOT1     (1-)      "   TOTAL NEW YORK"
    ROWL     (S1)      " NEW JERSEY"
                       "  LAKEWOOD"
    ROWM     (1)       "  MONTCLAIR"
    ROWN     (1)       "  NEWARK"
    TOT2     (1-)      "   TOTAL NEW JERSEY"
    TOT3     (S1==)    "   TOTAL EASTERN REGION"
```

These lines would print as follows:

```
  EASTERN REGION
   NEW YORK
    ALBANY                     xxx
    BUFFALO                    xxx
    CHAMPLAIN                  xxx
                              -----
     TOTAL NEW YORK            xxx

   NEW JERSEY
    LAKEWOOD                   xxx
    MONTCLAIR                  xxx
    NEWARK                     xxx
                              -----
     TOTAL NEW JERSEY          xxx

                              =====
     TOTAL EASTERN REGION      xxx
                              =====
```

ROW NAMES

The nine rows have the names ROWA, ROWB, ROWC, TOT1, etc. Names must begin with an alphabetic character ("A" through "Z") and may be as long as eight characters. The second and subsequent characters may be numeric ("0" though "9") as well as alphabetic. You may not use any other characters. You should use names that convey as much meaning as possible or that closely parallel existing account numbers, organization numbers, etc.

PAGE BREAKS

ROWA begins on a new page because of the "P" indicator on the first line. (Usually a "P" is not required on the first row, because the printer will already be positioned to a new page and the "P" would cause another page break.)

BLANK LINES

Between the lines "TOTAL NEW YORK" and "NEW JERSEY" there is a blank line because of the "S" (for space) indicator used in the ROWL specification. Also the final total "TOTAL EASTERN REGION" is separated from the "TOTAL NEW JERSEY" line by a blank line due to the "S" in the TOT3 specification.

OVERSCORE AND UNDERSCORE

The "TOTAL NEW YORK" and "TOTAL NEW JERSEY" lines are preceded by an overscore which separates the totals from their details. These are caused by the hyphens "-" appearing in the TOT1 and TOT2 specifications.

The "TOTAL EASTERN REGION" line is preceded by an overscore and followed by an underscore due to the equals signs "=" which appear in the TOT3 specification.

DECIMAL POINT SPECIFICATION

All lines will be printed with one decimal point because of the "1" which appears in each specification. Note that in reports where all data are to be printed with the same number of decimal points and the number of columns is small, then it may be easier to place the decimal point indicator in only the column specification. Note that this specification has no effect on the values within a row, only on the way that they appear.

SPECIFICATION PRECEDENCE

If there are decimal point specifications for both a row and a column then the larger of the two specified values will be used.

SUMMARY OF PRINT SPECIFICATIONS FOR ROWS

Each row specification may contain information which controls pagination, spacing, decimal point positioning, etc. Specification of this information occurs between parentheses "(" and ")" and is placed after the row name and before the quote mark which introduces descriptive headings. Between the parentheses may be any of the following:

"P" - indicates that a page break should occur before the row is printed.

"S" - indicates that a blank line should occur before the row is printed. Up to three "S" characters may be used. Each "S" will result in one blank line.

"T" - indicates that the row is generally a calculated or "total" row and all values may be set to zero using the "z" (zero) command in the display mode.

"X" - sometimes it is convenient to set aside temporary rows to hold intermediate calculation results. An "X" causes PLAN80 to ignore the row during printing. The row will, however, be displayed on the screen for possible confirmation of intermediate results.

"n" - a digit between "1" and "6" to specify the minimum number of positions to be printed to the right of a decimal point. When this parameter has been specified for both rows and columns the larger prevails. No decimal point will be printed for a value whose row and column specifications have no decimal point indicator.

"-" or "=" - you may input any two characters other than those noted above to print an over or underscore. Usually the characters would be a minus or an equals sign because they highlight effectively. However, periods, asterisks or blanks, etc., could be used. The first character always applies to an overscore, printed before the line, and the second to an underscore, printed after the line. (If you want only an underscore then a suitable character may be added to the specification of the following row.)

MULTIPLE LINE HEADINGS

Row descriptions and headings may occur on any number of lines, though each line must be enclosed in quotes and the total number of characters is limited to 256 for each row.

In the row specification any number of print lines may be specified on a single input line as in

```
  ROWX    "LINE 1"  " LINE 2"  "  LINE 3"
```

However, the technique of lining up the left-most row description quotes makes it easier to visualize the printed effect as in

```
  ROWX    "LINE 1"
          " LINE 2"
          "  LINE 3"
```

WHEN NO DESCRIPTION IS PROVIDED

The row name is used as a description if no description is provided. In other words, the following two row specifications are equivalent.

```
  ROW99
  ROW99      "ROS99"
```

PLACEMENT OF A ROWS SECTION

Normally a ROWS section will occur only once within a PLAN80 application. Furthermore, at least one row must be specified before any data may be entered or any calculations performed. A TITLES or OPTIONS section may precede any ROWS section, and a COLUMNS section must precedé a ROWS section.

If a second ROWS section is encountered then PLAN80 assumes you are starting a new application within the same physical run. This is permissable and is an effective way to handle complex runs which require no operator intervention. All rows, data and previously computed values will be forgotten. Any TITLES, COLUMNS and OPTIONS will survive the introduction of a new ROWS section, however.

CHAPTER 4

STARTING VALUES

A "DATA" SECTION

Data statements must occur in a data section headed

```
:DATA
```

Each data statement contains
- * a row or column name,
- * an equals sign "=", and
- * a series of numbers and/or certain other characters.

Each statement must begin on a new line, but statements may be continued on as many lines as necessary.

A PLAN80 application may have any number of data sections.

SOME EXAMPLES

```
:DATA
  SALES    =  100  120  ///  160
  ADMIN    =   16   18   (*1.08)
  INDEX    =  (+1)
  LONGONE  =  101  102  103  104
              105  106  107  108
```

ENTRY OF DATA

To discuss the entry of data into PLAN80 let us think of the columns and rows as determining a checkerboard, or matrix, as follows. Also think of each box, or cell, as having a value of zero before any data is entered.

```
           COLA    COLB    COLC
          --------------------
ROW1     |     0 |     0 |     0 |
         |--------------------|
ROW2     |     0 |     0 |     0 |
         |--------------------|
ROW3     |     0 |     0 |     0 |
         |--------------------|
ROW4     |     0 |     0 |     0 |
          --------------------
```

ROW OR COLUMN ENTRY

Data may be entered across any row or down any column. The method used is determined by the name used in the data statement.

If it is a row name, say ROW2 in the above example, then the data values that follow the name ROW2 will be assigned to COLA, COLB, etc., respectively. For instance, the data statement

```
ROW2 = 10 20 30
```

will cause the cell at the intersection of ROW2 and COLA to contain 10. The cell at COLB will contain 20, etc., as follows:

	COLA	COLB	COLC
ROW1	0	0	0
ROW2	10	20	30
ROW3	0	0	0
ROW4	0	0	0

Similarly a second data statement, coming after the first,

```
COLB = 40 50 60 70
```

would produce the following result:

	COLA	COLB	COLC
ROW1	0	40	0
ROW2	10	50	30
ROW3	0	60	0
ROW4	0	70	0

Note that the 50 of the second statement replaced the 20 of the first in the ROW2,COLB cell. Data are always assigned on a replacement basis in the order encountered.

LENGTH OF THE DATA STATEMENT

The data statement need not have the same number of data elements as the number of rows or columns. For instance, the statement

```
ROW4 = 100 200
```

in addition to the previous ones would result in

```
            COLA    COLB    COLC
          --------------------
ROW1      |     0 |   40 |     0 |
          |--------------------|
ROW2      |   10 |   50 |   30 |
          |--------------------|
ROW3      |     0 |   60 |     0 |
          |--------------------|
ROW4      |  100 |  200 |     0 |
          --------------------
```

Had the ROW4 statement contained four or more numbers as in

```
  ROW4  =  100  200  300  400
```

then the 300 would have been assigned to the ROW4,COLC position, but the 400 (and any subsequent numbers) would have been ignored.

DATA GENERATION

Often when data are input for a significant number of periods, the first few will represent "actual" experience or a firm "plan" and those following will be "estimated" or "projected" future values. Also the last absolute value is frequently used as a base to create subsequent projected values. For example, sales units for last year and this year's plan may be "hard" numbers, but subsequent year values are frequently constructed as an increase of a particular percentage or absolute amount over the last firm number. PLAN80 provides several shortcut methods for generating new data.

CHANGE BY A CONSTANT PERCENTAGE

As an example, let us assume that after several values you want to set the remaining values to increase by 10% over each prior period. The following would achieve this effect:

```
  ROWX  =  80  100  (*1.10)
```

If six columns had been defined then the above statement would be equivalent to:

```
  ROWX  =  80  100  110  121  133.1  146.41
```

Similarly a declining relationship may be produced by

```
  ROWX  =  100  (*.90)
```

which for four columns is equivalent to

```
ROWX  =  100  90  81  72.9
```

The above technique is useful for producing numbers which increase or decrease by a constant percentage over time. It is also possible to produce the same effect by division. For example, the statement

```
ROWX  =  100  (/1.10)
```

is equivalent to

```
ROWX  =  100  90.9091  82.6446  75.1315  etc.
```

CHANGE BY A CONSTANT AMOUNT

It is also possible to generate data values that change by a constant amount as shown by the following two examples:

```
ROWX  =  100  (+2)
ROWY  =   50  (-3)
```

which would be equivalent to

```
ROWX  =  100  102  104  106  108  etc.
ROWY  =   50   47   44   41   38  etc.
```

REPETITION

A value may be repeated using a slash "/" as in the following:

```
ROWX  =  10  20  ///
ROWY  =  10  ///  20
```

These statements are equivalent to:

```
ROWX  =  10  20  20  20  20
ROWY  =  10  10  10  10  20
```

A value may be repeated to the end of a row or column by using a plus "+" as in the following. The "+" must, however, be the last character of a data input statement:

```
ROWX  =  10  20  +
ROWY  =  10  +  20   <-- not permitted
```

The first of the above statements is equivalent to

```
ROWX  =  10  20  20  20  20  20  etc.
```

SKIPPING

Sometimes it is desirable to skip one or more rows or columns when entering data. Skipping may be done within an individual statement or may be controlled in general for groups of statements using the FOR statement.

For skipping in individual statements an asterisk "*" is used. Assume, for instance, that the first three rows had data that were not to be changed, but that you did wish to change data in rows four through seven of a particular column. This could be accomplished with the statement

```
  COLX  =  *  *  *  14  15  16  17
```

The asterisks "*" may occur anywhere that a number may occur. They have no value, unlike zero, which is a value and which would replace some other value. A statement of the form

```
  COLX  =  10  *  20  *  *  30  *  40
```

is valid. As with a statement containing only numeric values only as many values will be used as can fit into the row or column named at the beginning of the statement.

MIXING FEATURES

Most of the features discussed so far may be used together within a single data statement. For example, the following statements are valid (though they may be confusing and probably should be avoided):

```
  COLA  =  10  ///  (*1.10)
  ROW2  =  10  /  *  /  +
  ROW3  =  /  /  (+1)
```

These examples are equivalent to:

```
  COLA  =  10  10  10  10  11  etc.
  ROW2  =  10  10   *  10  10  etc.
  ROW3  =   0   0   1   2   3  etc.
```

In the first example the "///" repeats the 10 three times, and the "(*1.10)" functions as it does normally. The second demonstrates that a slash "/" repeats the most recent value even though one or more no-value asterisks "*" intervene. The third example demonstrates that an initial previous value of zero is assumed at the start of each statement.

USE OF THE "FOR" STATEMENT

It is also possible to control the entry of data using a FOR statement. The exact format of the FOR statement is covered elsewhere, but a discussion of what it does and how it works is appropriate at this point. In brief, the FOR statement makes PLAN80 behave as though one or more rows and/or columns did not exist. (A FOR statement, using the ALL parameter, is also used to make PLAN80 recognize those same rows or columns again.)

It is probably easiest to understand the FOR statement in terms of some examples. Let's use the checkerboard again:

	COLA	COLB	COLC
ROW1	0	0	0
ROW2	0	0	0
ROW3	0	0	0
ROW4	0	0	0

If the statement

```
:FOR COLUMNS = (COLA,COLC)
```

which omits COLB is encountered then PLAN80 becomes blind to COLB until another FOR statement recognizes COLB. The statement

```
ROW2 = 10 20 30
```

would result in the following:

	COLA	COLB	COLC
ROW1	0	0	0
ROW2	10	0	20
ROW3	0	0	0
ROW4	0	0	0

Note that the COLB position was skipped as expected, and that the 20 value was put into COLC, not the 30. This feature is useful in providing actual data to a particular column, the month of June, for example, and avoiding those for January through May.

DIFFERENT INPUT SCHEMES

We will look at five methods of inputting to the month of June. Often the FOR approach, covered last, is the most desirable.

METHOD 1

Input on a row basis, repeating actual data from previous months:

```
SALES  =  106  108  115   98  118  125
COSTS  =   60   60   62   58   64   68
ADMIN  =   10    9   11   12   10   12
 etc.
```

This method is very effective for smaller applications. Prior month data may be corrected easily. But the possibility of accidentally changing a prior month number might make this approach undesirable.

METHOD 2

Input on a row basis, skipping prior months:

```
SALES  =    *    *    *    *    *  125
COSTS  =    *    *    *    *    *   68
ADMIN  =    *    *    *    *    *   12
 etc.
```

This method is easy in January but awkward in December. You have to count the asterisks to know what month is being updated. Forgetting to add an asterisk each month would result in data being assigned to the wrong column.

METHOD 3

Input on a column basis:

```
JUNE  =  125  68  12  etc.
```

This is a very clean approach if data are readily available in the right sequence. However, if new rows ever need to be inserted this becomes a poor approach.

METHOD 4

Input on a row basis into a dummy column and then move the data from the dummy to the JUNE column:

```
  SALES  =  125
  COSTS  =   68
  ADMIN  =   12
   etc.
```

This is often a good approach, but it does introduce an extraneous column.

METHOD 5

Input on a row basis directly into the JUNE column:

```
  :FOR COLUMNS = (JUNE)
  :DATA
    SALES  =  125
    COSTS  =   68
    ADMIN  =   12
     etc.
  :FOR COLUMNS = ALL
```

This approach is frequently the best if Method 1 is not feasible. Note that if the second FOR statement is omitted a subsequent RULES or DATA section may be affected.

DATA FORMAT

Input values may contain a leading minus sign, a decimal point and/or a scale factor and a percent sign as in the following:

```
  ROWX  =  105  -11  -12.964%  -1.893E-16  .00306   2.92E18
```

You must be careful not to embed blanks, commas or any other characters other than those shown.

DATA PRECISION

The precision of a number (the number of significant digits) is a machine limitation, equal to six or seven digits for most micro-computer systems. For example, if the number 999,999,999 is entered (without the commas, of course) it will be treated as 1.000000 * 10^9 (read one times ten to the ninth power) which is off by one part in a billion.

This same number could be entered as 1.0E9 or 10E8, etc., where the number following the "E" is a power of ten. The power of ten may be negative as in 1.0E-6 to represent the very small number, 0.000001. Very few business problems will need the "E" scaling feature, but it is available for those who do need it. The maximum and minimum "E" values are machine dependent, usually +38 and -38 respectively for most microcomputers.

ON-LINE ENTRY OF VALUES

You may enter any number into any row/column location from the data input mode of the display mode, as described in chapter 6. There may be times, however, when you would like to have PLAN80 ask you to input one or more specific values as data statements are being processed. For example, you might have a series of calculations based on a sales assumption, or you might want to input an inflation rate for each iteration of a model. Anytime that it is permissable to use a number you may use a question mark, and PLAN80 will subsequently ask you to

```
  Input value:
```

The following examples show how you might use the question mark and how comments can be used.

```
  SALES    =  ? ? ?   : Input values for first three periods

  SALES    =  ?(*?)   : First period value and growth (eg. 110%)

  IntRate  =  12.8%  14.2%  ? +  :Provide rate for future periods
```

THE "INITIALIZE" STATEMENT

The format of the INITIALIZE statement is:

```
  :INITIALIZE
```

The INITIALIZE statement zeroes all values. PLAN80 always does this at the end of a ROWS section. However, you may process multiple sets of data within one application, using INITIALIZE to insure that no values remain from a prior set.

CHAPTER 5

CALCULATIONS

A "RULES" SECTION

Rules, or calculation statements, must occur within a rules section headed by

```
:RULES
```

Rules contain

- a row or column name,
- an equals sign "=", and
- a PLAN80 expression.

Each rule must begin on a new line but may continue on subsequent lines. The last character of a line to be continued must, however, be a mathematical operator (+,-,*,/ or ^).

A PLAN80 application may have any number of rules sections.

SOME EXAMPLES

The following are indicative of valid PLAN80 rules:

```
ROW14    =  ROW9  +  ROW13
ROW26    =  ROW16  *  @SUM(ROW17..ROW25)
PRODA    =  TOTAL - @MAX(DEMAND1,FACTOR3)
LINE5    =  @SIN(@LN(LINE3))
TWOLINE  =  ROW66 + ROW76 -
              31 * (ROW23 - ROW18)
```

ROW RULES AND COLUMN RULES

To understand just how rules operate in PLAN80 let us use a checkerboard with the following initial values:

	COLA	COLB	COLC
ROW1	15	10	0
ROW2	40	30	0
ROW3	50	25	0
ROW4	0	0	0

If ROW4 was meant to be the sum of ROW1, ROW2 and ROW3 we could write the rule

```
ROW4  =  ROW1  +  ROW2  +  ROW3
```

which would calculate a total of 105 for COLA, 65 for COLB and 0 for COLC. Similarly, we could write the rule

```
  COLC  =  COLA  -  COLB
```

which would calculate 5 for ROW1, 10 for ROW2, 25 for ROW3 and 40 for ROW4, based on the results of the first calculation.

Note that the first rule is a row rule and that it contains no column names. Similarly the second rule is a column rule, and it contains no row names. Except when referring to the specific square on the checkerboard, as in @(ROW4,COLC), the names used in valid PLAN80 expressions must all be of the same type as the name preceding the equals sign. Row and colımn rules may be freely intermixed as we did above, but any single rule will have only column names or only row names.

Also you should note that a row rule applies to each and every column unless excluded by a FOR statement. Similarly column rules apply to each and every row unless excluded by a FOR statement. *The FOR statement discussed in detail in Chapter 8.*

SOME BASIC EXAMPLES

Let's continue the discussion of rules with another example which has data in ROW1 only:

	COLA	COLB	COLC
ROW1	2	4	7
ROW2	0	0	0
ROW3	0	0	0
ROW4	0	0	0

and consider the following rules:

```
  ROW2  =  3
  ROW3  =  ROW1  *  ROW2
  ROW4  =  ROW1  *  3
```

NOTE: The first example looks like a data statement, but don't confuse data and rules statements.

The above rules would result in the following:

	COLA	COLB	COLC
ROW1	2	4	7
ROW2	3	3	3
ROW3	6	12	21
ROW4	6	12	21

The first rule replaces all values of ROW2 with the value 3. This is equivalent to the data statement

```
ROW2  =  3  3  3
```

The second rule creates ROW3 by multiplying each value of ROW1 by the corresponding value in ROW2.

The third rule creates the same effect as the two previous rules combined. Each value within ROW1 is multiplied by the number 3.

RULES ARE NOT EQUATIONS

It is permissable to use the row or column named as the answer in an expression on the right side of a rule. The rule

```
ROW1  =  ROW1  *  3
```

would cause each value within ROW1 to be replaced by a value three times the original value, and

```
ROW1  =  ROW1  +  ROW1  +  ROW1
```

would be a cumbersome but permissable way of achieving the same result.

MATHEMATICAL OPERATORS

Rules may have names, single values such as "3", and expressions linked by the mathematical operators (+,-,*,/ and ^ for addition, subtraction, multiplication, division and exponentiation). Note that the asterisk "*" and not the character "x" is used for multiplication. The "^" indicates that one value is to be raised to the power indicated by another. Thus ROW3^2 indicates that each value of ROW3 is to be multiplied by itself and ROW3^.5 indicates the square root of each ROW3 value.

IF . . . THEN . . . ELSE Logic

PLAN80 permits rules of the form

```
  IF <condition> THEN <statement> ELSE <statement>
```

and

```
  IF <condition>
    THEN <statement>
    THEN <statement>
    ELSE <statement>
    ELSE <statement>
```

The following are valid rules

```
        IF ROWA<0 THEN ROWA = 0
        IF ROWA<ROWB THEN ROWX = ROWA ELSE ROWX = ROWB
        IF @(ROW2,COL7)>200 THEN COL8 = COL11
        IF ( (A>B) OR (C<>D) ) AND (E<F)
          THEN X=0
          THEN Y=0
          THEN Z=0
```

Notes on IF . . . THEN . . . ELSE . . .

a. IF, THEN and ELSE may be in upper or lower case as with all PLAN80 statements.

b. Comparisons in a condition phrase may use the following operators:

=	-	equals
<>	-	not equal
<	-	less than
>	-	greater than
<=	-	less than or equal to
>=	-	greater than or equal to

c. Equals and not equal comparisons can produce unexpected results. For instance two values, A and B, may appear as 100, but one may really be 99.6 rounded up for display purposes. To test equivalence it is generally preferable to compare the difference between the absolute value of two numbers with an appropriate limit. For instance, when no decimal points are used A and B can be treated as equal when they appear equal with

```
     IF @ABS(A-B)<1.0 THEN ...
```

d. Conditions may be compounded using AND and OR, and parentheses may be used to control the order of evaluation. In the absence of parentheses conditions and statements are evaluated from left to right. WARNING - This is different from most programming languages which use more complex evaluation rules.

e. If the comparison in a condition is between rows (e.g. ROWA>ROWB) then statements following a THEN or ELSE must begin with a row name (as in ROW21 = ...). Similarly for columns. You may, however, use conditions involving only single values (as in @(R2,C3)<18) with either row or column statements.

f. Any number of THEN's or ELSE's may follow an IF, and they may be in any order, and simple statements may occur between THEN and ELSE statements as in

```
IF A>B
  THEN C=A
  B=G+H+I
  ELSE C=B
```

g. Sometimes it is prefereable to use functions such as @MAX, @MIN or @LOOKUP instead of IF ... THEN ... ELSE ... . For example, the following rules

```
ROWA = @MAX(ROWA,0)
ROWX = @MIN(ROWA,ROWB)
```

are equivalent to

```
IF ROWA<0 THEN ROWA = 0
IF ROWA<ROWB THEN ROWX = ROWA ELSE ROWX = ROWB
```

ORDER OF EVALUATION

The expressions on the right side of a rule are evaluated from left to right. Thus

```
  ROW1  =  3  +  6  /  2
```

would result in the assignment of 4.5 to each column of ROW1. Parentheses may be used to alter the evaluation sequence as in

```
  ROW1  =  3 + (6  /  2)
```

which would result in the assignment of 6 to each ROW1 value. PLAN80 always determines the value of an expression within parentheses before using it as a term of some other expression.

In PLAN80 any number of parentheses may be used to control the order of calculations within a rule. You must insure, however, that, starting from the left, the cumulative number of closing parentheses is never greater than the number of opening ones.

MATHEMATICAL FUNCTIONS

In addition to the normal calculations involving values, row and column names and expressions within parentheses, PLAN80 provides a number of powerful mathematical functions. They are summarized on the following page and are described in detail in the following paragraphs. Note that all are prefixed with the "@" sign.

@SUM

This function is used to add groups of rows or columns. For example, the rule

```
  ROW9  =  @SUM(ROW1,ROW4..ROW7)
```

is equivalent to

```
  ROW9  =  ROW1  +  ROW4  +  ROW5  +  ROW6  +  ROW7
```

assuming that ROW5 and ROW6, and only those rows, occur between ROW4 and ROW7.

In general any number of single names (as in "ROW1" above) or group expressions (as in "ROW4..ROW7") may be used within the @SUM function.

Mathematical Functions in PLAN80

@SUM	- calculates the sum of groups of rows or columns
@AVG	- calculates averages
@CUM	- calculates the cumulative sum within a row or column
@MAX	- determines the largest of a list of values
@MIN	- determines the smallest of a list of values
@ABS	- determines the absolute value
@SQRT	- determines the square root
@INT	- determines the integer portion of a number
@FRAC	- determines the fractional portion of a number
@EXP	- determines "e" (2.71828) raised to the value provided
@LN	- determines the natural logarithm
@LOG	- determines the log (base 10)
@SIN	- determines the trigonometric
@COS	- functions of angles
@TAN	- expressed in radians
@ASIN	- determines the trigonometric arcfunctions
@ACOS	- (e.g., the angle in radians whose
@ATAN	- function is the value provided)
@LOOKUP	- establishes values based on table lookup
@IRR	- calculates the internal rate of return (e.g., the discount rate at which a series of values discounted over time has a present value of zero)
@SL	- straight line depreciation
@SOD	- sum of digits depreciation
@DB	- declining balance depreciation
@DBXSL	- declining balance switch to straight line depreciation

---

In the @SUM function only row or column names may occur, but the @SUM function may appear within expressions as in the following:

```
ROW9   =  @SUM(ROW1..ROW8) * 100
ROW99  =  @SUM(ROW1..ROW9) + @SUM(ROW51..ROW59)
ROW99  =  100  -  @SUM(ROW91..ROW98)
```

@AVG

The average function is identical in format to the @SUM function. The value produced equals the sum divided by the total number of rows or columns included in the specification. For instance, the following example would result in the sum's division by 4.

```
ROW99  =  @AVG(ROW1..ROW3,ROW7)
```

@CUM

This function is used to calculate the cumulative sum of values occurring within a row or column. Thus, assuming data in ROWQ of 10, 14 and 26, the rule

```
  ANSWER  =  @CUM(ROWQ)
```

would result in row ANSWER's having values of 10, 24 and 50.

@MAX

This function determines the largest of a series of values and is used in the form

```
  ANSWER  =  @MAX(exp1,exp2...expn)
```

where expn is a row or column name or an expression enclosed in parentheses. Assume the following values:

	COLA	COLB	COLC
ROW1	5	10	15
ROW2	10	0	-5
ROW3	8	5	3

The following rules would result in the values shown:

```
  ROWX  =  @MAX(ROW1,ROW2)           =   10   10   15
  ROWY  =  @MAX(0,ROW2)              =   10    0    0
  ROWZ  =  @MAX((ROW1-ROW2),ROW3)    =    8   10   20
```

@MIN

This function has the same format and can be used in the same situations as @MAX, except that it determines the minimum of the values provided.

The function @MIN(0,ROWX) may be used with @MAX to separate negative and positive values within a row or column. For example, if ROWX has the values -2, 0 and 10 then

```
  ROWA  =  @MAX(0,ROWX)  will result in   0   0  10  and
  ROWB  =  @MIN(0,ROWX)  will result in  -2   0   0
```

@LOOKUP

The lookup function is used for problems where it is difficult or impossible to develop a mathematical expression which relates the value of one number to another. For example, consider a set of income tax rates. As income increases from one bracket to another, new rates become applicable. Other examples would include interest rates as a function of the size of a loan, prices as a function of quantity purchased, and overhead rate as a function of cost center number.

The lookup function has three parts as in the following example:

```
  ANSWER  =  @LOOKUP(INCOME,BRACKET1..BRACKET3,RATE1)
```

The first part is INCOME in the example. Where it falls in the range of BRACKET values (the second part) will determine which RATE is selected (the third part). Consider the following:

	COLA	COLB	COLC	COLD
INCOME	2	6	20	100
BRACKET1	5	5	5	5
BRACKET2	12	12	12	12
BRACKET3	30	30	30	30
RATE1	.08	.08	.08	.08
RATE2	.12	.12	.12	.12
RATE3	.33	.33	.33	.33
ANSWER	0	.08	.12	.33

For each column the row ANSWER has been determined by searching the rows from BRACKET1 to BRACKET3 for the last row whose value is less than or equal to the value of INCOME. Then ANSWER is set to the value of the row whose rate value corresponds to the bracket value. For instance, in COLB the value of INCOME is greater than the BRACKET1 value, which causes the RATE1 value to be assigned to ANSWER. In COLC the income value first equals or exceeds the BRACKET2 value, and the RATE2 value is assigned to ANSWER. Similarly in COLD, ANSWER receives the RATE3 value. (If there is no row with a value less than INCOME, as in COLA, then ANSWER is set to zero).

The following guidelines, expressed in terms of the example above, must be followed in setting up lookup tables:

* If ANSWER is a row then the other three parts, income, brackets and rates, must be rows. ANSWER could, however, be a column, but then the other parts must be columns.
* The brackets and rates tables may occur anywhere. Either table may precede the other.
* Within each table, however, all member rows or columns must be stored next to one another. In other words, no row, not part of the brackets table, may occur between BRACKET1 and BRACKET3. Otherwise the row would be considered part of the bracket table. Similarly, no row, not part of the rates table may occur between RATE1 and RATE3.
* You must make sure that the rate table is the same length as the brackets table. Table length is implied by the number of rows or columns occurring between and including BRACKET1 and BRACKET3.
* In the example above the row names for brackets and rates contained numbers. This is a useful technique, but not a necessary one. Any valid name may be used for table rows or columns.

@IRR

This function is generally used to evaluate the projected cash flows of alternative investments as in

```
ANSWER = @IRR(CASHFLOW)
```

The internal rate of return is calculated separately as of each period for which cumulative cash flow is positive. The following example assumes an initial investment of 100 and a return of 50 each year thereafter. The internal rate of return is zero for years 1-3, 23.4% for years 1-4, 34.8% for years 1-5, etc.

	Year1	Year2	Year3	Year4	Year5	Year6	Year7
CASHFLOW	-100	50	50	50	50	50	50
ANSWER	-	-	-	23.4	34.8	40.9	46.6

This function can be explained as follows: Investments are generally characterized by an initial outlay and a subsequent return of cash. Obviously the greater the total amount returned and the earlier it is returned, the more attractive the investment. But how should two investments be evaluated if one provides larger returns per dollar invested, but at a later date?

The @IRR function determines the discount rate that makes the present value of future earnings equal to the value of current outlays. Now, the higher that future earnings are without any change in timing, the higher the discount rate must be raised to balance future earnings with current outlays. Thus the higher the calculated internal rate of return, the more attractive the investment, all other factors being equal.

The @IRR function calculates the discount rate "r" in the following series where "cf(n)" indicates the cash flow of year "n":

```
  0  =  cf(1)  +  cf(2)/(1+r)  +  cf(3)/(1+r)^2  +  . . .
```

Note: the @IRR function calculates annual rates, but you may use other periods (months, quarters, etc.) and then annualize the rates. The following will convert from monthly to annual rates:

```
  ROWY = ROWX / 100 + 1 ^ 12 - 1 * 100
```

DEPRECIATION

PLAN80 calculates the following types of depreciation:

* straight line,
* sum of digits,
* declining balance, and
* declining balance switch to straight line.

The following examples show how each type operates. Assume there is a row named COST and assume the following rules:

```
  SL    = @SL(COST,6,1)
  SOD   = @SOD(COST,6)
  DB2   = @DB(COST,6,2.0,1)
  DB15  = @DB(COST,6,1.5,1)
  DBXSL = @DBXSL(COST,6,2.0,1.0)
```

The results (in terms of accumulated depreciation) would be

	Year1	Year2	Year3	Year4	Year5	Year6	Year7	Year8
COST	1000							
SL	167	333	500	667	833	1000	1000	1000
SOD	286	524	714	857	952	1000	1000	1000
DB2	333	556	704	802	868	912	941	961
DB15	250	438	578	684	763	822	867	900
DBXSL	333	556	704	857	952	1000	1000	1000

All methods require that you indicate a row (or column) which contains the amount to be depreciated (COST) and provide an expected life (6 years in the examples). The declining balance methods also require a factor (2.0 for double declining, 1.5 for 150% declining, etc.). And the straight line and declining balance methods require that you provide a fraction of the first year for which depreciation is to be taken (1 for assets acquired at the beginning of a year, 0.5 for those using a mid-year convention, etc.).

The row COST contains 1000 in the first year, and the other rows show cumulative depreciation as of each period. It is possible to place a new cost amount in every column of the COST row. Each result lines (SL,SOD, etc.) would show the applicable cumulative depreciation for costs in the current and previous years.

OTHER MATHEMATICAL FUNCTIONS

The input data and functions below produce the results shown:

-- INPUT DATA --				-------	RESULTS	---)----
COLA	COLB	COLC	Function	COLA	COLB	COLC
-30	-	20	@ABS	30.00000	-	20.00000
100	200	300	@SQRT	10.00000	14.14214	17.32051
-1.5	0.5	10.2	@INT	-1.00000	-	10.00000
-1.5	0.5	10.2	@FRAC	-0.50000	0.50000	0.20000
0.5	1	2	@EXP	1.64872	2.71828	7.38906
1	10	100	@LN	-	2.30258	4.60517
0.1	1	10	@LOG	-1.00000	-	1.00000
0.5	1	2	@SIN	0.47943	0.84147	0.90930
0.5	1	2	@COS	0.87758	0.54030	0.41615
0.5	1	2	@TAN	0.54630	1.55741	2.18504
0.5	1	2	@ASIN	0.52360	1.57080	-
0.5	1	2	@ACOS	1.04720	-	-
0.5	1	2	@ATAN	0.46365	0.78540	1.10715

SINGLE DATA VALUES

You may use single data values in any expression where you might use a number. Single data values are formed as follows

```
@(row name,column name)
```

The following would be valid rules using single data values:

```
COL14  =  COL13/@(ROW99,COL13)
ROW6   =  @MAX(ROW4,@(ROW5,COL11))
```

ASSIGNMENT OF SINGLE VALUES

It is possible to assign values to a single checkerboard location as in the following:

```
(ROW2,COL3)  =  12.3
(ROW6,COL9)  =  @(ROW3,COL11)
```

Note that the term to the left of the equals sign does jot use the "@" sign. Also note that expression on the right of the equals sign must reduce to a single value. Thus the following would be invalid:

```
  (ROW3,COL5) = ROW9
```

LEADING MINUS SIGNS

Any term appearing in an expression, including row or column names, data values or functions, may be preceded by a minus sign to indicate that the quantity represented is to be treated as though it were to be multiplied by minus one. For example, the following reflect permissable use of leading minus signs:

```
  ROWA  =  -ROWB
  ROWA  =  -(ROWC * -ROWD)
  COLA  =  -@MAX(COLF,COLG)
```

SHIFTING OF DATA

Consider a rule such as

```
  SALES  = UNITS  *  PRICE
```

If this is a row rule then every column will be treated independently of all other columns. We want most rules to act this way. But there are some problems where we want to bring a value forward from a prior period and add it to or subtract it from a value of the current period. This requires a row rule that can operate on more than one column at a time. Shifting is what makes this possible in PLAN80.

SHIFTING IN ROW RULES

First, let's look at PLAN80's ability to shift data, starting with the following:

	COLA	COLB	COLC	COLD
ROW1	11	22	33	44
ROW2	0	0	0	0
ROW3	0	0	0	0
ROW4	0	0	0	0

Within a row rule PLAN80 allows data to be moved from one column to another by adding a bracketed number at the end of any term or expression. The number provides the left or right direction, using the checkerboard analogy, for the source of data values. For example, look at the effects of the following rules:

```
ROW2  =  ROW1(-1)
ROW3  =  ROW1(-2)
ROW4  =  ROW1(1)
```

	COLA	COLB	COLC	COLD
ROW1	11	22	33	44
ROW2	0	11	22	33
ROW3	0	0	11	22
ROW4	22	33	44	0

SHIFTING IN COLUMN RULES

Data may be similarly shifted between rows by column rules. For example, assume the following:

	COLA	COLB	COLC	COLD
ROW1	11	0	0	0
ROW2	22	0	0	0
ROW3	33	0	0	0
ROW4	44	0	0	0

The following rules would produce the results shown:

```
COLB  =  COLA(-1)
COLC  =  COLA(-2)
COLD  =  COLA(1)
```

	COLA	COLB	COLC	COLD
ROW1	11	0	0	22
ROW2	22	11	0	33
ROW3	33	22	11	44
ROW4	44	33	22	0

OTHER WAYS TO VIEW SHIFTING

It may be helpful to consider a calculation such as

```
  ROW2  =  ROW1
```

as equivalent to

```
  ROW2  =  ROW1(0)
```

and to consider a calculation such as

```
  ROW2  =  ROW1(-1)
```

as equivalent to asking that PLAN80 move data from each previous column in transferring values from ROW1 to ROW2.

Another way of viewing the shift instruction is in terms of time. Assuming time progresses from COLA to COLB to COLC, etc., then

```
  ROWX(-1)    indicates one period ago,
  ROWX        indicates the present, and
  ROWX(1)     indicates one period in the future.
```

THREE EXAMPLES OF SHIFTING

Shifting is used frequently for cash flow and inventory problems because balances at the end of one period depend on those of a prior period. We will look at two cash flow and an inventory problem.

CASH FLOW EXAMPLE #1

Let's start with the following:

	Y1980	Y1981	Y1982	Y1983	Y1984
INCOME		30	35	30	40
ASSETS	50	60	68	62	70
LIAB	20	24	27	35	30
CF1					
CF2					
CF3					

We will assume income is not complicated by taxes and that the ASSET and LIAB lines represent single lines from a balance sheet. You could think of them as inventories and accounts payable. In lines CF1, CF2 and CF3 we will show the cash flow from each of the first three lines.

CURRENT PERIOD CHANGES

Without taxes to worry about we can view income as a direct cash inflow. Thus

```
  CF1  =  INCOME
```

reflects the fact that during each period one part of the total cash flow effect will be the that of income received.

CHANGES BETWEEN PERIODS

Calculating the cash flow effect for balance sheet items is more difficult than for an item such as income, because we are interested in changes between two periods rather than the activity of one period. If, for example, an asset increases from one period to the next, then cash will be required to acquire this asset. Adopting the convention that cash outflows should be negative on the cash flow statement, the following rule may be used to calculate the cash flow effect of asset changes:

```
  CF2  =  ASSET(-1) - ASSET
```

The liability account is handled in a similar way, only the sign must be reversed. When a liability increases you are increasing a debt to someone and this debt is a source of cash. Thus you could use the following for a liablity:

```
  CF3  =  LIAB  -  LIAB(-1)
```

We may now look at the effect of these three rules:

	Y1980	Y1981	Y1982	Y1983	Y1984
INCOME	-	30	35	30	40
ASSETS	50	60	68	62	70
LIAB	20	24	27	35	30
CF1	-	30	35	30	40
CF2	-	(10)	(8)	6	(8)
CF3	-	4	3	8	(5)

Note that we have put no values in the first column since the ASSET(-1) and LIAB(-1) terms are undefined for that column.

CASH FLOW EXAMPLE #2

In the first example we assumed income had already been determined for each period before calculating cash flow. In this example we will let income for each period depend on the cash balance as of the end of the previous period. This means that we will now need to calculate both the income and cash flow for one period before calculating them for the next.

Assume the following:

	Y80	Y81	Y82	Y83	Y84
RATE		.20	.15	.10	.10
REVENUE		200	240	300	350
EXPENSE		120	140	163	180
INCOME1					
INCOME2					
CASHBAL	100				

The problem may be handled in either of two ways.

In the first approach you should break the problem into two sections. First, handle in general terms all calculations which do not depend on prior year values. In the example only one rule is required for this

```
  INCOME1 = REVENUE - EXPENSE
```

Then continue as follows:

```
  (INCOME2,Y81) = @(CASHBAL,Y80) * @(RATE,Y81)
  (CASHBAL,Y81) = @(CASHBAL,Y80) + @(INCOME1,Y81) + @(INCOME2,Y81)
  (INCOME2,Y82) = @(CASHBAL,Y81) * @(RATE,Y82)
  (CASHBAL,Y82) = @(CASHBAL,Y81) + @(INCOME1,Y82) + @(INCOME2,Y82)
  (INCOME2,Y83) = @(CASHBAL,Y82) * @(RATE,Y83)
  (CASHBAL,Y83) = @(CASHBAL,Y82) + @(INCOME1,Y83) + @(INCOME2,Y83)
  (INCOME2,Y84) = @(CASHBAL,Y83) * @(RATE,Y84)
  (CASHBAL,Y84) = @(CASHBAL,Y83) + @(INCOME1,Y84) + @(INCOME2,Y84)
```

This will result in the following:

```
                 Y80     Y81     Y82     Y83     Y84
              ----------------------------------
 RATE         |     - |  .20 |  .15 |  .10 |  .10 |
              |----------------------------------|
 REVENUE      |     - |  200 |  240 |  300 |  350 |
              |----------------------------------|
 EXPENSE      |     - |  120 |  140 |  163 |  180 |
              |----------------------------------|
 INCOME1      |     - |   80 |  100 |  137 |  170 |
              |----------------------------------|
 INCOME2      |     - |   20 |   30 |   33 |   50 |
              |----------------------------------|
 CASHBAL      |   100 |  200 |  330 |  500 |  720 |
              ----------------------------------
```

An alternative approach uses other PLAN80 facilities. First, set up a file which has

```
  :RULES
    INCOME1 = REVENUE - EXPENSE
    INCOME2 = CASHBAL(-1) * RATE
    CASHBAL = CASHBAL(-1) + INCOME1 + INCOME2
```

Save this file as "CASHCALC". Then create a main procedure which includes CASHCALC for each period as in

```
  :COLUMNS
    <columns>
  :ROWS
    <rows>
  :DATA
    <data>
  :FOR COLUMNS=(Y81)
  :INCLUDE "CASHCALC"
  :FOR COLUMNS=(Y82)
  :INCLUDE "CASHCALC"
  :FOR COLUMNS=(Y83)
  :INCLUDE "CASHCALC"
  :FOR COLUMNS=(Y84)
  :INCLUDE "CASHCALC"
  :DISPLAY
```

The FOR and INCLUDE statements are covered in Chapter 8. The FOR statement controls the scope of the rules, telling PLAN80 to process one year at a time in the above example. The INCLUDE statement tells PLAN80 to fetch a group of statements and process them as though they had been part of the file containing the INCLUDE statement.

INVENTORY EXAMPLE

Let us now look at how we might construct a projected inventory position, a process that is somewhat the reverse of the cash flow example. In the cash flow problem we knew the projected balances and were interested in calculating the changes. In this problem the changes, production and shipments, will be known and we will be interested in computing the balances.

Let's begin with the following values:

	MAY	JUN	JUL	AUG
OPENBAL	100			
PROD	20	25	30	30
SHIP	25	20	24	29
CLOSEBAL				

Now if we were doing the above problem manually we might calculate the MAY closing balance of 95 as with

```
  CLOSFRAL  =  OPENBAL  +  PROD  -  SHIP
```

and then use a rule such as

```
  OPENBAL  =  CLOSEBAL(-1)
```

to carry the 95 forward to the next period and repeat the process. This, however, involves switching back and forth between rules starting with MAY and then moving forward to JUN, JUL, etc. However, it is important that you understand that each row rule operates on all columns before moving to the next rule, and that each column rule operates on all rows before carrying out the next rule. Thus, to use this approach you must use the INCLUDE technique introduced in the second cash flow example.

There is, however, an easier way to handle the inventory problem. The solution is to put the net changes into the CLOSEBAL line and accumulate them as follows:

```
  CLOSEBAL  =  OPENBAL  +  PROD  - SHIP
  CLOSEBAL  =  @CUM(CLOSEBAL)
```

The first rule would result in the following:

	MAY	JUN	JUL	AUG
OPENBAL	100			
PROD	20	25	30	30
SHIP	25	20	24	29
CLOSEBAL	95	5	6	1

and the second rule would change the last row to

```
CLOSEBAL      95     100     106     107
```

Finally the rule

```
OPENBAL  =  OPENBAL  +  CLOSEBAL(-1)
```

would complete the example:

	MAY	JUN	JUL	AUG
OPENBAL	100	95	100	106
PROD	20	25	30	30
SHIP	25	20	24	29
CLOSEBAL	95	100	106	107

RELATIVE SHIFTING

The shift examples above indicate "absolute" shifting. "Absolute" means that a term such as ROWX(-1) always refers back one column. In the presence of subtotals, however, it is often desireable to refer back a variable number of rows (or columns). Consider, for example, the calculation of cash receipts as a function of prior period sales. You might have a rule

```
RECEIPTS = SALES(-1)
```

which would work if all months were next to one another as in

```
JAN   FEB   MAR  APR   MAY   JUN    . . . etc.
```

However, if quarterly subtotals were introduced then the calculation would not be correct. Instead of the "absolute" reference to the prior column SALES(-1) you may use the FOR statement to limit the scope of calculation results to only the months (and not the quarterly subtotals) and may use "relative" shifts to skip over the excluded columns. Use of the FOR statement is discussed in detail in Chapter 8. Relative shifts

However, if quarterly subtotals were introduced then the calculation would not be correct. Instead of the "absolute" reference to the prior column SALES(-1) you may use the FOR statement to limit the scope of calculation results to only the months (and not the quarterly subtotals) and may use "relative" shifts to skip over the excluded columns. Use of the FOR statement is discussed in detail in Chapter 8. Relative shifts are indicated by adding an ampersand "&" at the end of the shift value as in SALES(-1&). Thus the introduction of quarterly subtotals could be handled using

```
:FOR COLUMNS=(JAN..MAR,APR..JUN,JUL..SEP,OCT..DEC)
:RULES
  RECEIPTS = SALES(-1&)
```

The final effect, before calculating quaterly subtotal values, might be as follows:

	JAN	FEB	MAR	QTR1	APR	MAY	JUN	QTR2 ... etc.
SALES	100	110	120		130	140	150	
RECEIPTS		100	110		120	130	140	

USE OF PERCENT SIGN IN NUMBERS

In RULES statements you may use a percent sign at the end of any number to indicate that it is to be considered a percent.

ON-LINE ENTRY OF VALUES

Whenever you use a question mark in place of a number PLAN80 will ask you to

```
Input value:
```

You may use this feature in RULES statements as in

```
:RULES
  SALES    =  ADV(?)      : Input lag in months as in "-2"
  OVERHEAD =  SALES * ?   : Input rate as in "8%"
```

CHAPTER 6

OUTPUT STATEMENTS

DISPLAY AND PRINT - AN INTRODUCTORY NOTE

DISPLAY provides an interactive environment in which you may scroll through the output on a computer screen, add or change data, and request printed output if and when desired. You would normally use the DISPLAY approach whenever you were viewing results on the screen, changing assumptions, etc., and then printing a set of results.

PRINT is available for more structured jobs, such as for standard analyses performed each month, or where many printed reports are desired. Unlike DISPLAY, PRINT does not operate interactively. Through use of the OPTIONS feature,however, you have nearly the same control over printed output as you have in the DISPLAY mode.

THE "DISPLAY" STATEMENT

When the DISPLAY command is encountered PLAN80 lets you view in tabular and graphic form values from any part of the matrix formed by all rows and columns. You can also print a polished report on your screen or printer (or on disk) or you can input new values and ask PLAN80 to recalculate results.

CONTROLLING WHAT IS DISPLAYED

The computer screen may be thought of as a window which may look at any part of the matrix formed by all columns and rows. Using commands shown on the bottom of your screen you may position the window at any particular row and/or column and may scroll through the results in any direction.

CONTROLLING HOW IT IS DISPLAYED

You may also change the width of columns or the effective size of the screen or print page. And you have the option of having rows and/or columns identified by their names instead of their descriptions. These choices allow you to display the greatest possible amount of information on the screen at one time.

DATA DISPLAY MODE

Initially the computer screen will be positioned to show the upper left corner of your model, and the value for the first row and column will be highlighted. At the bottom of your screen is a reminder of which keys you can use to control the display

(SPACEBAR,ESCape,h,i,I,j,J,k,K,m,M,p)

These keys operate as follows

SPACEBAR

The spacebar takes you to the data input mode, described subsequently. You will also use the spacebar to return to the display mode.

ESCape

The escape key takes you to the display options mode, described subsequently, which lets you change display characteristics, print a report, enter the graphic analysis mode, recalculate values, etc.

H - Home

The H key (upper or lower case) always returns the highlighted position on the screen to the upper left corner of the display if it is not already at that position. If it is already there then keying h will redisplay the model starting at its first row and column.

MOVE UP - comma,i,I

Keying lowercase i or a comma causes the highlighted position to move up one row. If you precede the key with a number (as in 12i) the highlighted position will move up more then one row at a time (12 in this case). If the new position to be highlighted is not already on the screen then the screen will be redisplayed starting with the row containing the highlighted value.

Keying upper case I causes the highlighted position to move to the top of the screen if it is not already there. When the highlighting is at the top of the screen I causes redisplay, and the highlighting moves to the first row of the model.

MOVE LEFT - < .j.J

These keys operate similarly to the i and I keys, except that they cause the highlighted position to move to the left. Lower case j and the < key may be preceded by a number to cause multiple columns to be skipped. Upper case J will move the highlighted position to the first column of the screen and then to the first column of the model.

MOVE RIGHT - >.k.K

These keys are similar to the j and J keys except that they cause movement to the right.

MOVE OOWN - period. m.M

These keys are similar to the i and I keys except that they cause downward movement.

P - Page

The upper or lower case P key always causes a redisplay of the screen and the highlighted position is always placed at the upper left corner of the screen.

Scrolling

Using combinations of the above keys you can scroll though the results of your model. For instance, to scroll down a model when it is first displayed you need only press upper case M to place the highlighted position at the bottom of the screen and P (upper or lower case) to cause the page to be redisplayed. The last row of the previous screen will become the first row of the new screen. This process can be repeated any number of times until you reach the bottom of your model.

To scroll upwards you need only press 16i to move upwards 16 rows at a time. Similarly you might use the keys KP in succession to scroll from left to right and 6j to move from right to left.

DATA INPUT MODE

As noted above you use the spacebar to enter and leave the data input mode. The data input mode is always identified by the message

```
  Input Value:
```

at the bottom of the screen. You may input one value at a time or you may generate a series of values.

Single Values

The following are examples of valid values:

```
  1   88.9   -333   40%   -82.3%
```

To input a single value you may key in a value and then terminate it with a carriage return, the spacebar or with any of the cursor movement/redisplay keys (h,i,j,k,m,p - upper or lower case). A spacebar or RETURN will cause you to leave the input mode. The other keys will move the highlighted position as described above and will leave you inside the data input mode.

Computed Values

Any value for input may be a computed value as in the following

```
  10463*.9973
  294+8%
```

Any two numbers may be linked by +,-,* or / signs, but the expression may have no spaces (since the spacebar would terminate the entry).

Recalling the Last Value

You may start any specification of a value with a R (upper or lower case) to recall the last value input. This may be done after any number of intervening cursor moves. For example, if the last entry had been 100/.90 (or 111.111) then you could move the highlighted position anywhere and enter the same value again simply as R or r. Similarly r could be used as part of another calculation as in r/.9 (or 123.457). Note that, if used, the r must be the first character of the entry.

GENERATING MULTIPLE VALUES

To generate multiple values you may add a colon and a repeat factor to any value, even to a computed one. You must terminate your entry with an i,j,k,m key to indicate the direction in which generated values are to be placed. The following examples show how you might generate indices, inflation assumptions, sales projections, costs, etc.

entry	result				
13:4k	13	13	13	13	
100*108%:5k	108	117	126	136	147
100k (followed by)					
r*108%:4k	100	108	117	126	136
1980+1:5k	1981	1982	1983	1984	1985
1*.85:4k	.850	.723	.614	.522	

DISPLAY OPTIONS MODE

From the display mode you may enter the options mode by pressing the ESCape key. You will see at the bottom of your screen

```
Input Code -->
(SPACEBAR(for help),ESCape,1,2,3,4,5,6,7,8,s,d,g,p,w,x,z,+,!,#)
```

If you press the spacebar you will see the following

```
DISPLAY OPTIONS

1 - Column Heading Option  . descriptions
2 - Row Heading Option . . . descriptions
3 - Row Description Width  .  18
4 - Column Width . . . . . .   8
5 - DISPLAY Page Width . . .  80
6 - DISPLAY Page Length  . .  24
7 - PRINT Page Width . . . .  80
8 - PRINT Page Length  . . .  56
S - Set Standard Options
D - Display
G - Graph
P - Print
W - Write a DATA file
X - Exit from DISPLAY
+ - Recalculate
! - Quit Current File
# - Exit from PLAN80

Input Code -->
```

The following paragraphs explain the purpose and use of the above codes.

1 - COLUMN HEADING OPTION

On either the screen or printed report, columns may be identified by their three-line descriptions or by column names. The former is the standard option. The option may be changed by keying 1. If the Display Options screen is showing you will see the display change from "descriptions" to "names" or vice versa. If you are in the display mode the screen will be redisplayed with the other option prevailing. The names option allows you to see the names used and provides two more lines for display of data.

2 - ROW HEADING OPTION

On either the screen or printed report, rows may be identified by the row description (which may be up to 255 characters on any number of lines) or by names (up to 8 characters on one line). The former is the standard option. The option may be changed by keying 2. The options display will change from "descriptions" to "names" or vice versa. The row descriptions width will automatically be set to 18 for descriptions and 9 for names. From the display mode the screen will be redisplayed. Using names generally allows another column and several more rows to be shown on the screen.

3 - ROW DESCRIPTION WIDTH

In printing or displaying lines such as

```
  SALES            100   200    . . . etc.
  COST OF SALES     60   120    . . . etc.
```

PLAN80 places the words "SALES" AND "COST OF SALES" in a special column whose width is determined by this option. No matter how long the description line, only the number of characters, including blanks, given by this option will be shown. Thus, in reviewing intermediate results you may want to change this parameter from its standard value of 18 to 9, for example, to display an additional column of numbers. The width may be changed by keying "3" and responding to the prompt line

```
  Input Row Description Width -->
```

with any number between 0 and 40.

4 - COLUMN WIDTH

In printing or displaying results the standard width of columns is 8 characters. For numbers without decimal points this permits any value less than 10 million to be displayed, and for others it allows six significant digits to be shown. The width option may be changed by keying "4" and responding to the prompt line

```
  Input Column Width -->
```

with any number between 4 and 20.

5 - DISPLAY PAGE WIDTH

The standard value of this parameter is 80, the width of most computer screens. The value may be changed by keying "5" and responding to the prompt line

```
  Input DISPLAY Page Width  -->
```

with any number between 40 and 132.

6 - DISPLAY PAGE LENGTH

The standard value of this parameter is 24. The value may be changed by keying "6" and responding to the prompt line

```
  Input DISPLAY Page Length  -->
```

with any number between 6 and 99.

7 - PRINT PAGE WIDTH

The standard value of this parameter is 80. The value may be changed by keying "7" and responding to the prompt line

```
  Input PRINT Page Width  -->
```

with any number between 20 and the number of characters which your printer is able to put on a single line (up to 240).

8 - PRINT PAGE LENGTH

The standard value of this parameter is 56 which will provide 5 lines at the top and bottom of a 11" page at six lines per inch. The value may be changed by keying "8" and responding to the prompt line

```
  Input PRINT Page Length  -->
```

with any number between 6 and 999.

S - SET STANDARD OPTIONS

Key "S" to set all options to their standard values.

D - DISPLAY

From the DISPLAY OPTIONS screen you must press D to return to the display mode.

G - GRAPHIC ANALYSIS

Keying G lets you enter the graphic analysis mode which is described subsequently.

P - PRINT

Key P to put results on an external printer, on your screen or in a text file on disk for subsequent editing and/or printing. Printed results include titles, underlining and spacing which is omitted when results are displayed. You will be asked

```
  INPUT Destination -->
```

You may enter the name of an output device such as "PRINTER:" or "CONSOLE:", or you may enter a file name such as "REPORT14.TXT" or "A:OUTPUT" and your report will be written to disk.

If you enter the name of an existing file it will be deleted. You should pick file names with care.

When printing to disk you will see and hear nothing for a few seconds. Then your disk will go on and off. This pattern will repeat several times for a short report and may require several minutes for a long report.

If you press only the RETURN key the output will automatically go to your screen. You may want to do this to see how your printed report will appear before actually printing it.

If any error is detected, such as no more space on a diskette then you will hear a warning bell and subsequent output will appear on your console.

At the end of each page printed on your screen a message will appear asking you to press the spacebar. Output to your printer is continuous.

You may press the escape key to leave this option without printing anything.

The columns and rows shown and the width of columns, etc., are controlled by the options whose values are shown on the DISPLAY OPTIONS screen. ¯When results are printed you should insure that the highlighted position is at the first row and column which you wish to print. Columns and rows preceding the starting values, as represented by the cross-hatched area in the following diagram, will not be printed. (You may press the h key twice to go to the "home" position.)

```
            COLA      COLB    . . .  COL20
           --------------------------------
  ROW1     |/ / / |/ / / / / / / / / |
           | / /  | / / / / / / / /  |
  ROW2     |/ / / |/ / / / / / / / / |
           |-------------------------|
  ROW3     | / /  | xx.xx            |
    .      |/ / / |   ^              |
    .      |/ / / |   |              |
    .      | / /  |  highlighted     |
           |/ / / |  position        |
  ROW99    |/ / / |                  |
           --------------------------------
```

In the above example any rows or columns preceding the highlighted position would not be shown, but the remaining ones would be divided into as many physical pages as necessary, as explained in the PRINT section. By moving the highlighted position to the appropriate location before printing you may print summary sections (which tend to be at the end of models) without printing all the details.

W - WRITE DATA STATEMENTS

Pressing W causes data statements to be created. As with a printed report the file created may be placed on disk for subsequent editing or for use by any PLAN80 model which uses the same row and/or column names that appear on the DATA statements. Generally this feature will be used to save assumptions after a session of creating data interactively in the data input mode.

Z - ZERO CALCULATED ROWS ANO COLUMNS

Key Z to zero all values in rows and/or columns whose definitions include a "T" (to indicate that their values are to be calculated rather than input).

X - EXIT FROM "OISPLAY"

Keying "X" causes PLAN80 to leave the DISPLAY mode and to continue with the first statement following the DISPLAY command.

+ - RECALCULATE

If you press "+" PLAN80 will return to the most recent INTERACTIVE statement and continue from that point (usually at the beginning of a RULES section). If there is no INTERACTIVE statement then this key will cause PLAN80 to start again at the beginning of the current file.

! - LEAVE CURRENT FILE

Keying "!" causes PLAN80 to leave the current procedure file. You normally use this option when you want to leave your current application and run another one.

# - EXIT FROM PLAN80

If you key "#" you will leave PLAN80 and will see the CP/M command line.

MISCELLANEOUS

Each DISPLAY option value remains in effect until changed from the DISPLAY menu or through an OPTIONS command or until you leave the PLAN80 system.

GRAPHIC ANALYSIS

```
                       PLAN80 Graphic Analysis
     1000 +                                           title
Y .       |                                           type scatter
- .       |                                 1         "y" axis
A .       |                             1                row1  SALES
X .       |                         1                    row2  NET
I     800 +                     1                        row3  none
S A       |                 1                            upper none
  N       |             1                   2            lower none
H D       |         1                                    heading
E         |     1                                       "x" axis
A C   600 +1                            2                first F1981
D O       |                                              last  F1989
I N       |                         2                    columns
N T       |                                              upper none
G U       |                     2                        lower none
  A   400 +                                              heading
  T       |                 2
  I       |             2
  O       |         2
  N       |2    2
      200 +--------+-------+--------+-------+-------+
                            Last Column Shown: F1989
(ESCape,D-isplay,SPACEBAR,<,>)
```

The above example shows a typical display from the graphic analysis mode.

CONTROL IN THE GRAPHIC ANALYSIS MODE

Within the graphic analysis mode you may control what variables are displayed and how they are displayed using the 14 items that appear along the right hand column. The purpose and use of each item is discussed subsequently.

Each item must be be selected before it can be used for control. To select an item you must move the highlighted position to the appropriate line (using the < and > keys) and then press the spacebar. How to change control values is discussed on the following pages. When you are satisfied with the control values you may press the spacebar again (and use the < and > keys to move the highlighted position again or you may press D to display the graph.

The following line appears at the bottom of your screen.

```
(ESCape,D-isplay,SPACEBAR,<,>)
```

ESCape

Press the escape key to leave the graphic analysis mode.

D-ISPLAY

You may press the D key at any time to display a graph whose appearance will be governed by the control values indicated along the richt side of your screen.

< , >

These keys (upper and lower case) move the highlighted line up and down the list of control features. You may enter a number as in 4>, for example, to move by more than one step at a time.

SPACEBAR

Once you have highlighted the feature of interest (using the < and > keys) you select a feature using the spacebar. Each feature is discussed in the following paragraphs.

TITLE

The title feature lets you change the title which is displayed at the top of the screen. If you select "title" the existing title line will disappear and you may enter up tc 60 characters. If you enter a new line of zero length then the previous title will be restored.

TYPE

Each time you press the spacebar when the word "type" is highlighted the type will advance through its four available types, scatter, #1 bar, #2 bar, and cumulative.

A scatter graph was shown at the beginning of this section. In it the value of each row selected was plotted in the y-axis against each column in the x-axis in the order that the columns occurred. That type of a graph shows how values vary over time.

In a scatter graph which plots row values in the y-axis against row values in the x-axis you can see how one row appears to depend of the values in another (as in net income vs. interest rate, for example).

A cumulative graph is similar to the row vs. column scatter graph. The value plotted as point 2 in each column is, however the sum or the row1 and row2 values. Similarly the value plotted as point 3 is the sum of row1, row2 and row3 values.

The following diagrams show the difference between the two bar graph types.

```
|   #1 bar                      |   #2 bar
|                               |
|                               |
|     2 1                       |   1       2
|  2  2 12 1                    |   11     222
|  2 12 12 12                   |  111     2222
| 12 12 12 12                   | 1111     2222
| 12 12 12 12                   | 1111     2222
| 12 12 12 12 etc               | 1111 etc 2222 etc
 -----------------               ------------------
```

ROW 1, ROW 2, ROW 3

When you select any one of these three lines the current row name or the word "none" will be highlighted. Each time that you press the > key the next row name will appear, starting from the first row of your model. You may move back towards the first row using the < key. You may precede either of these keys with a number to advance multiple rows at a time. You can select any one, two or three rows at a time, but you won't be able to set all three of these items to "none".

UPPER AND LOWER BOUNDS (Y-AXIS)

When data values are graphed data is automatically scaled to fill the entire y-axis. You may choose to override this feature by specifying an upper or lower bound. Once you select a bound you may increase it by the equivalent of one line on the y-axis for each time that the < key is pressed, or decrease it by a similar amount for each time the > key is pressed. Whenever a bound has been selected you may return it to the automatic mode by pressing the / key, and "none" will appear by the word "upper" or "lower". When you use bounds you must remember to turn them off when they are no longer needed.

HEADING (Y-AXIS)

Selecting this item lets you provide a two column heading for the y-axis. The display at the bottom of your screen indicates with a + sign the end of each column.

FIRST, LAST COLUMN

These two items will show the names of the first and last column names occurring in a model. Sometimes you will want, however, to restrict the scope of columns analyzed (excluding total columns, for instance). You may do this by selecting one of these items and selecting a first or last column name as in the row selection process.

COLUMN/ROW (X-AXIS)

This item applies only to the "scatter" type of graph. Other types always assume a selection of "column". For scatter graphs you may choose to have the x-axis comprised of columns (between the first and last column) or you may set it to a row whose values will be plotted against the values of rows in the y-axis.

When you initially select this item a highlighted asterisk will appear next to the word "columns". Then if you press the > key you will see the word "row" followed by the first row name, just as occurs in the y-axis selection of rows. You may input a number before the < or > keys to move rapidly to the row of interest.

UPPER, LOWER BOUNDS (X-AXIS)

Selection and adjustment of the x-axis bounds is only meaningful for type "scatter". Otherwise these bounds function similar to those of the y-axis. Each increment of adjustment (caused by the < and > keys) is equal to one column on the graph (2% of the difference between the original upper and lower bounds). Striking / removes a selected bound.

HEADING (X-AXIS)

A heading for the x-axis may be entered by selecting the last item and entering characters as for the titles at the top of the graph.

THE "PRINT" STATEMENT

Printed output may be obtained by using the "P" option of the DISPLAY command or from using the PRINT command as in

```
  :PRINT (<limits>) "<output destination>"
```

EXAMPLES

```
  :PRINT
  :PRINT "FILE21.TXT"
  :PRINT "B:FILE4.TXT"
  :PRINT (10,23,2,6)
  :PRINT (SALES,NET,P01,P10) "PRINTER:"
  :PRINT (1,999,F1981,F1983)
```

LIMITS

The limits field is optional and indicates the first and last rows and columns which are to appear on a printed report. All rows and all columns will be printed if you do not use this field. The format of this field is (r1,r2,c1,c2) where r1 and r2 represent the first and last rows and c1 and c2 the first and last columns to be printed. If used the field must contain four values within parentheses in the order shown. Each row or column indicator may be either a name or number as in the examples above. If you use an invalid name or number r1 will be set to the first row, r2 to the last row, c1 to the first and c2 to the last column.

OUTPUT DESTINATION

The output destination is also optional and if used must be in quotes. It may be the name of an output device such as "PRINTER:" or "CONSOLE:", or it may be a file name such as "REPORT14.TXT" or "B:OUTFILE.TXT" and your report will be written to disk. If not provided the output destination will be CONSOLE:.

If you provide a file name without indicating the drive then your output will be placed on the currently logged on disk. If you enter the name of an existing file then it will be deleted before your new file is created. You should pick printfile names with care to avoid destroying files accidentally.

When printing to disk you will see and hear nothing for a few seconds. Then your disk will go on and off. This pattern will repeat several times for a short report and may require several minutes for a long report.

MULTIPLE PAGE PRINTOUTS

When reports are printed PLAN80 automatically formats them into as many pages as are required to show all columns and rows (except those excluded because of a "X" in their print control field). For instance, an application using 20 columns and 50 rows might require three physical pages to handle the width and two to handle the length, or a total of six pages. The order of the pages would be as shown in the following diagram:

	Columns 1-8	Columns 9-16	Columns 17-20
Rows 1-30	Page 1	Page 2	Page 3
Rows 31-50	Page 4	Page 5	Page 6

Each printed page includes all row and column descriptions applicable for that page.

In applications with large numbers of rows and columns you have the option of specifying where page breaks will occur or PLAN80 will automatically create them using prevailing values for row description width, column width and page width options.

Standard assumptions for PLAN80 are 18 characters for row description width and 8 characters for column width. Thus, a page, 80 characters wide, would hold 7 data columns. If column width were reduced to 7 or 6 characters then the total number of columns which would fit on the page would increase to 8 and 10 respectively.

When you are using the PRINT statement nonstandard values for column width, page width, page length, etc., are introduced using the OPTIONS statement.

THE "OPTIONS" STATEMENT

The OPTIONS statement includes the word OPTIONS and one or more parameters which may be separated by blanks or line breaks as in

```
:OPTIONS STANDARD
```

or

```
:OPTIONS
  PAGEWID(100)
  PAGELEN(60)
```

OPTIONS parameters override standard values for row description width, column width, page length, etc. Many option values may be changed in the DISPLAY mode, but OPTIONS allows you to control output appearance without operator intervention.

OPTION PARAMETERS

STANDARD - Sets the following parameters to their standard values.

COLNAMES - Sets column description option to print one line names instead of three line descriptions.

COLDESC - Sets column description option to print three line descriptions. This is the standard option.

ROWNAMES - Sets row description option to print names instead of descriptions.

ROWDESC - Sets row description option to print descriptions instead of names. This is the standard option.

ROWWID(n) - Sets row description width to n as long as n is between 0 and 40. Standard value is 18.

COLWID(n) - Sets column width to n as long as n is between 4 and 20. Standard value is 8.

PAGELEN(n)- Sets page length to n as long as n is between 6 and 999. Standard value is 56.

PAGEWID(n)- Sets page width to n as long as n is between 40 and 240. Standard value is 80.

BLANK
DASH
ZERO - Controls how zero values will be shown. Standard value is DASH.

PLACEMENT OF AN OPTIONS SECTION

An OPTIONS section may be placed anywhere in a stream of PLAN80 statements. Options chosen remain in effect until changed in a subsequent OPTIONS section or by choices made in the Display Options mode.

CHAPTER 7

COMMUNICATING BETWEEN APPLICATIONS

PUT AND GET - AN INTRODUCTORY NOTE

PUT and GET statements are used to communicate values from one PLAN80 application to another. The PUT statement causes results to be written to disk and the FOR statement controls just what portion of the results are saved. The GET statement causes saved results to be included, and the FOR statement controls how saved values are assigned.

THE "PUT" STATEMENT

The PUT statement has one parameter, a file name, which must be in quotes.

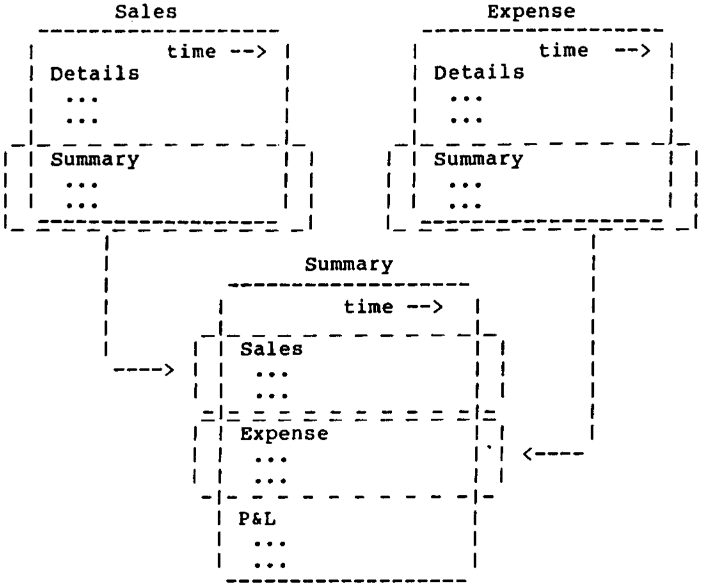
```
:PUT "<filename>"
```

The PUT statement saves PLAN80 results on disk for use by another PLAN80 application. For example, one application may create a sales analysis by product and save the results of each sales district for subsequent preparation of a summary report. A similar approach might be used to consolidate financial statements or cost center budget and expense data.

The following examples should clarify the PUT and GET process. In this example all Unit A and Unit B values are saved and subsequently consolidated.

```
           Unit A                              Unit B
   ------------------------            ------------------------
 |        COLA  COLB  TOTC |          |        COLA  COLB  TOTC |
 | ROW1     10    20    30 |          | ROW1      5     8    13 |
 | ROW2     15    25    40 |          | ROW2     12    16    28 |
 | TOTR     25    45    70 |          | TOTR     17    24    41 |
   ------------------------            ------------------------
                    |                          |
                    |                          |
                    |                          |
                    V      Consolidated        V
                  ------------------------
                |        COLA  COLB  TOTC |
                | ROW1     15    28    43 |
                | ROW2     27    41    68 |
                | TOTR     42    69   111 |
                  ------------------------
```

In the following example one PLAN80 planning application is used for sales and another for expense. Summary results from each are then used by a third application for summarizing the plan:



In this example only a few values from each application are transferred to the summary.

Through PUT and GET statements PLAN80 allows you to send any portion of the values of one application to another.

USE OF THE "FOR" STATEMENT

The FOR statement, whose construction is explained in Chapter 8, controls which rows and/or columns are communicated between PLAN80 applications.

To understand more thoroughly what the FOR statement does let us consider the following example:

```
          COLA   COLB   COLC   COLD   COLE
        ----------------------------------
ROW1    |      |      |      |      |      |
        |----------------------------------|
ROW2    |      |      |      |      |      |
        |----------------------------------|
ROW3    |      |      |      |      |      |
        |----------------------------------|
ROW4    |      |      |      |      |      |
        |----------------------------------|
ROW5    |      |      |      |      |      |
        ----------------------------------
```

The FOR statement allows any number of rows or columns to be ignored during the PUT process. Those columns and rows not ignored appear in the same order that originally prevailed. Thus, if COLE, ROW1 and ROW3 were ignored, then the following rows and columns would be included.

```
          COLA   COLB   COLC   COLD
        ---------------------------
ROW2    |      |      |      |      |
        |---------------------------
ROW4    |      |      |      |      |
        |---------------------------
ROW5    |      |      |      |      |
        ---------------------------
```

Note that the data saved looks just as though it came from an application that was somewhat smaller than the original, and that the remaining values are in the same order as they were originally. It is as though you cut out strips representing rows and columns and slid the remaining pieces together.

THE "GET" STATEMENT

The format of the GET statement is

```
:GET "<filename>" mode
```

The name of the file that PLAN80 is to get must be in quotes and it must have been saved previously by a PUT statement.

The mode is optional and may be any of the following: REPLACE, ADD, SUBTRACT, MULTIPLY or DIVIDE. No quotes should be used. The default value is REPLACE, which means that any value assigned will replace any existing value. In a consolidation you would specify ADD so that each new value would be added to any existing value.

The other modes are useful in solving certain complex modeling problems. For example, MULTIPLY may be used for scaling standard sets of values, or patterns of 0 and 1 may be used as switches in turning various alternatives off or on.

HOW "GET" GETS

Getting data is somewhat more complicated than putting, because the number of columns or rows saved previously may be different from the number of rows and columns of the application containing a GET statement. PLAN80 handles this situation by ignoring values that do not "fit" in the current column and row structure. The following examples should clarify what is meant by "fit".

Assume that the PLAN80 application which contains the GET statement has the following columns and rows:

```
           COL1  COL2  COL3
          -------------------
 ROW1    |     |     |     |
         |-------------------|
 ROW2    |     |     |     |
         |-------------------|
 ROW3    |     |     |     |
         |-------------------|
 ROW4    |     |     |     |
          -------------------
```

First assume saved data of

```
           COLA
          ------
 ROWA    |     |
          ------
```

In this case the single value would go to the cell bounded by COL1 and ROW1 since they both occur first. Also there would be no effect on any value other than the one at COL1 and ROW1.

If the saved values were of the form

```
           COLA  COLB  COLC  COLD
          -------------------------
 ROWA    |     |     |     |     |
          -------------------------
```

then ROWA would be assigned to ROW1, because ROW1 is the first available row. COLA through COLC would be assigned to COL1 through COL3. COLD would be ignored because the number of columns in the saved data is greater than the number of columns in the current application.

Similarly with saved data of the form

	COLA	COLB
ROWA		
ROWB		
ROWC		
ROWD		
ROWE		

COLA and COL1 would correspond as would COLB and COL2. ROWA through ROWD would correspond to ROW1 and ROW4, and both values in ROWE would be ignored. In the assignment process there would be no effect on any value in COL3.

In each of the above examples the ROWA,COLA value was assigned to the ROW1,COL1 cell and other column and row values were assigned in the order of their occurance.

USE OF THE "FOR" STATEMENT

Just as the FOR statement can make the sending application (the one with the PUT statement) appear smaller than it actually is, it can do the same to the receiving application. Once the FOR statement has reduced the apparent size of an application, the assignment of data occurs in exactly the same manner as with the examples discussed above.

CHAPTER 8

OTHER STATEMENTS

THE "INTERACTIVE" STATEMENT

The INTERACTIVE statement is used in conjuction with the recalculate option of the DISPLAY statement. From the Display Options mode you may key + for recalculation, which will cause PLAN80 to continue processing at the first statement following the most recent INTERACTIVE statement.

This feature lets you set up a model using the following scheme:

```
:titles
 ..
:columns
 ..
:rows
 ..
:data
 ..
:interactive
:rules
 ..
:display
```

Control proceeds from the first statement to the DISPLAY statement at the end. You may view results, change assumptions, and then recalculate with control resuming at the RULES statement following the INTERACTIVE statement.

THE "INCLUDE" STATEMENT

The format of the INCLUDE statement is:

```
:INCLUDE "<filename>"
```

The filename must be the name of a file containing valid PLAN80 statements, and it must be in quotes. The filename may indicate a drive as in "A:MODEL1" or "B:MODEL1.TXT". The type designator "TXT" is optional, but will be appended if not already present.

The INCLUDE statement allows PLAN80 statements to be placed in a file external to an application and is useful when:

* a common set of column or row definitions are used for a number of applications,
* alternative sets of data assumptions have been created and are to be processed by one application,
* calculations must be repeated several times within a single application or in more than one application, or
* long production runs, involving many PLAN80 applications, are to run without operator intervention.

The first statement of an included file must be a control statement. It may be preceeded only by comment statements.

There is no restriction on the number of files which one file may cause to be included. For example, one file could call a number of others as follows:

```
: --- MONTHLY SUMMARIES ---
:INCLUDE "JOB01.TXT"
:INCLUDE "JOB02.TXT"
:INCLUDE "JOB99.TXT"
```

In addition each of the included files could include other files. For example, JOB01 might be constructed as follows:

```
: --- "JOB01.TXT" ---
:INCLUDE "COL01.TXT"
:INCLUDE "ROW01.TXT"
:INCLUDE "DAT01.TXT"
:INCLUDE "RUL01.TXT"
:DISPLAY
```

Generally you will find that even the most sophisticated applications rarely require more than two levels of included files, as in the examples above. Each level reduces the maximum possible size of an applicaton.

You must make sure that an included file never tries to include a file which caused it to be included. Circular references will bring PLAN80 to a halt and you will have to press the RESET key before correcting your PLAN80 input.

THE "REPEAT" STATEMENT

The format of the REPEAT statement is identical to that of the INCLUDE statement. The only difference is that REPEAT will cause a file to be included continuously until the ! key is used from the Display Options mode.

THE "FOR" STATEMENT

FOR statements begin with the word FOR and include a column and/or row limiter. Both limiters are optional and may occur in either order. Either may start on a new line or appear on the same line as the FOR.

EXAMPLES

```
:FOR COLUMNS=ALL ROWS=ALL
:FOR COLUMNS=(COLA..COLF,COLH)
:FOR ROWS=(ROW1,ROW3,ROW5)
:FOR COLUMNS=ALL  ROWS=ALL EXCEPT (ROW8..ROW12)
:FOR COLUMNS=(COLA..COLX) EXCEPT (COLG,COLP)
     ROWS=(ROW1..ROW17) EXCEPT (ROW7..ROW9)
```

COLUMN AND ROW LIMITERS

Limiters have the following format:

```
COLUMNS = (included columns) EXCEPT (excluded columns)
ROWS = (included rows) EXCEPT (excluded rows)
```

The EXCEPT and enumeration of excluded columns or rows are optional. In place of the enumeration of included columns or rows may be the word ALL which is not placed in parentheses.

In enumerating the included or excluded columns and rows you may either specify individual names as in

```
(ROW1,ROW2,ROW3,ROW4,ROW9)
```

and you may specify ranges as in

```
(ROW1..ROW4,ROW9)
```

The previous two examples are equivalent if ROW2 and ROW3 are the only rows occurring between ROW1 and ROW4. Individual names may appear in any order, but they must be separated by commas.

USE OF "FOR" STATEMENTS

The principles of the FOR statement have been covered in the chapters covering DATA, RULES, PUT and GET statements. You may find it of value to study those sections again. The purpose of FOR statements is to cause PLAN80 to ignore selected columns and/or rows.

You can enter data into a specific row or column as in

```
:FOR COLUMNS=(JUNE)
:DATA
  SALES  =  268.3
  COSTS  =  121.9
```

where SALES and COSTS are row names. Also the FOR statement might cause quarterly subtotal columns to be skipped as in

```
:FOR COLUMNS=(JAN..MAR,APR..JUN,JUL..SEP,OCT..DEC)
:DATA
  SALES  =  100  110  108  117   98   etc.
  COSTS  =   60   66   65   70   59   etc.
```

FOR statements may be used in the RULES mode to limit the action of calculations to a subset of the columns or rows. For example, assume actual sales data for two years is input directly and that future year sales data are calculated from values in other rows.

```
:DATA
  SALES  =  100  110
   ...
   ...

:FOR COLUMNS=(YEAR3..YEAR6)
:RULES
  SALES  =  UNITS  *  PRICE
   ...
   ...
```

Without the FOR statement the actual values 100 and 110 would become undefined, forcing you to input them after the future year calculations.

With PUT and GET it is easiest to think of the FOR statement as excluding columns and/or rows prior to saving or recalling values. PLAN80 then behaves, in both the PUT and GET modes, as though the actual application had been defined originally without the excluded columns or rows.

A REMINDER

Once columns or rows have been ignored because of one FOR statement, they will be ignored indefinitely until another recognizes them. Thus, you must generally include a second statement as in

```
:FOR COLUMNS=ALL
```

or

```
:FOR COLUMNS=ALL  ROWS=ALL
```

APPENDIX A

PLAN80 EXAMPLES

EXAMPLE 2

You should observe how names have been developed. Logical structure is important. You can duplicate blocks of row statements and rules, making only a few changes instead of typing all characters. Future changes will be easier and you will be more accurate.

Example 2 also shows how comments in the RULES section can be used effectively.

Also the OPTIONS feature is used to provide a nonstandard value for row description width. Lastly you should note how column headings have been structured.

```
                         PLAN80 EXAMPLE #2
                  Projected Sales and Gross Margin

                              ACT     EST    ------- PROJ -------
                             1981    1982    1983    1984    1985

TOTAL MARKET UNITS (MILLIONS)
 Product 1                    110     115     120     124     129
 Product 2                     88     103     118     136     157
 Product 3                     28      41      51      64      80
                            -----   -----   -----   -----   -----
  TOTAL                       226     259     289     325     366

ABC SHARE (PERCENT)
 Product 1                   30.0    31.0    32.0    33.5    35.0
 Product 2                   15.0    18.0    19.5    21.0    22.5
 Product 3                   20.0    20.0    19.0    18.0    17.0
                            -----   -----   -----   -----   -----
  TOTAL                      22.9    24.1    24.6    25.2    25.7

ABC UNITS (MILLIONS)
 Product 1                   33.0    35.7    38.3    41.7    45.3
 Product 2                   13.2    18.5    23.1    28.6    35.2
 Product 3                    5.6     8.2     9.7    11.5    13.6
                            -----   -----   -----   -----   -----
  TOTAL                      51.8    62.4    71.1    81.8    94.1

AVERAGE UNIT PRICE
 Product 1                  0.885   0.903   0.993   1.093   1.202
 Product 2                  0.510   0.550   0.594   0.642   0.693
 Product 3                  1.300   1.450   1.624   1.819   2.037
                            -----   -----   -----   -----   -----
  AVERAGE                   0.834   0.870   0.950   1.037   1.132

SALES DOLLARS (MILLIONS)
 Product 1                   29.2    32.2    38.0    45.5    54.4
 Product 2                    6.7    10.2    13.7    18.4    24.4
 Product 3                    7.3    11.9    15.8    21.0    27.7
                            -----   -----   -----   -----   -----
  TOTAL                      43.2    54.3    67.5    84.9   106.6

AVERAGE UNIT COST
 Product 1                  0.502   0.558   0.608   0.663   0.723
 Product 2                  0.317   0.310   0.285   0.262   0.241
 Product 3                  0.704   0.664   0.697   0.732   0.769
                            -----   -----   -----   -----   -----
  TOTAL                     0.477   0.498   0.515   0.533   0.549

AVERATE GROSS MARGIN (PERCENT)
 Product 1                   43.3    38.2    38.8    39.3    39.9
 Product 2                   37.8    43.6    52.0    59.1    65.2
 Product 3                   45.8    54.2    57.1    59.8    62.3
                            -----   -----   -----   -----   -----
  AVERAGE                    42.9    42.7    45.7    48.7    51.5
```

```
:TITLES
  1  "PLAN80 EXAMPLE #2"
  2  "Projected Sales and Gross Margin"
:COLUMNS
  F1981          "     ACT" "1981"
  F1982          "     EST" "1982"
  F1983          "    ----" "1983"
  F1984          "--- PROJ" "1984"
  F1985          " -------" "1985"
:ROWS
  MARKET1                   "TOTAL MARKET UNITS (MILLIONS)"
                            " Product 1"
  MARKET2                   " Product 2"
  MARKET3                   " Product 3"
  MARKET99   (-)            "  TOTAL"
  SHARE1     (S1)           "ABC SHARE (PERCENT)"
                            " Product 1"
  SHARE2     (1)            " Product 2"
  SHARE3     (1)            " Product 3"
  SHARE99    (1-)           "  TOTAL"
  UNITS1     (S1)           "ABC UNITS (MILLIONS)"
                            " Product 1"
  UNITS2     (1)            " Product 2"
  UNITS3     (1)            " Product 3"
  UNITS99    (1-)           "  TOTAL"
  PRICE1     (S3)           "AVERAGE UNIT PRICE"
                            " Product 1"
  PRICE2     (3)            " Product 2"
  PRICE3     (3)            " Product 3"
  PRICE99    (3-)           "  AVERAGE"
  DOLLAR1    (S1)           "SALES DOLLARS (MILLIONS)"
                            " Product 1"
  DOLLAR2    (1)            " Product 2"
  DOLLAR3    (1)            " Product 3"
  DOLLAR99   (1-)           "  TOTAL"
  COST1      (S3)           "AVERAGE UNIT COST"
                            " Product 1"
  COST2      (3)            " Product 2"
  COST3      (3)            " Product 3"
  COST99     (3-)           "  TOTAL"
  AGM1       (S1)           "AVERATE GROSS MARGIN (PERCENT)"
                            " Product 1"
  AGM2       (1)            " Product 2"
  AGM3       (1)            " Product 3"
  AGM99      (1-)           "  AVERAGE"
```

```
:DATA
  MARKET1 = 110 115 (*1.04)
  MARKET2 =  88 103 (*1.15)
  MARKET3 =  28  41 (*1.25)
  SHARE1  = .30 .31 .32 .335 .35
  SHARE2  = .15 .18 (+.015)
  SHARE3  = .20 .20 .19 .18 .17
  PRICE1  = .885 .903 (*1.10)
  PRICE2  = .510 .550 (*1.08)
  PRICE3  = 1.30 1.45 (*1.12)
  COST1   = .502 .558 (*1.09)
  COST2   = .317 .310 (*0.92)
  COST3   = .704 .664 (*1.05)
:RULES
  : Total Units
    MARKET99 = @SUM(MARKET1..MARKET3)
  : ABC Units = Total Units * Share
    UNITS1   = MARKET1 * SHARE1
    UNITS2   = MARKET2 * SHARE2
    UNITS3   = MARKET3 * SHARE3
    UNITS99  = @SUM(UNITS1..UNITS3)
  : Total Unit Share
    SHARE99  = UNITS99 / MARKET99 * 100
    SHARE1   = SHARE1 * 100
    SHARE2   = SHARE2 * 100
    SHARE3   = SHARE3 * 100
  : Sales Dollars = Units * Price
    DOLLAR1  = UNITS1 * PRICE1
    DOLLAR2  = UNITS2 * PRICE2
    DOLLAR3  = UNITS3 * PRICE3
    DOLLAR99 = @SUM(DOLLAR1..DOLLAR3)
  : Average Price = Dollars / Units
    PRICE99  = DOLLAR99 / UNITS99
  : Cost = Units * Unit Cost
    AGM1     = UNITS1 * COST1
    AGM2     = UNITS2 * COST2
    AGM3     = UNITS3 * COST3
    AGM99    = @SUM(AGM1..AGM3)
  : Average Cost = Cost / Units
    COST99   = AGM99 / UNITS99
  : Gross Margin = Marginal Revenue / Sales
    AGM1     = DOLLAR1-AGM1 / DOLLAR1 * 100
    AGM2     = DOLLAR2-AGM2 / DOLLAR2 * 100
    AGM3     = DOLLAR3-AGM3 / DOLLAR3 * 100
    AGM99    = DOLLAR99-AGM99 / DOLLAR99 * 100
:OPTIONS
  ROWWID(30)
:DISPLAY
```

EXAMPLE 3

Example 3 shows how a cash flow statement may be derived from an income statment and a balance sheet. You should note the introduction of the decimal point specification in the column statements. Also observe how important underlining, spacing, pagination and indentation are to the effective presentation of results, particularly if the report is to be presented to someone less familiar than yourself with the numbers and structure of your problem.

The rules of this example introduce the @CUM function, the shift feature and the FOR statement. The latter prevents calculations from affecting the 1980 column of the cash flow page. This is done since cash flow values are generally constructed as the difference between values for consecutive years, and there are no prior year values for the 1980 column.

The general logic of the cash flow rules is:

1. Calculate the income statement values (R02-R08).
2. Calculate the details of current assets (R22-R27).
3. Calculate details of liability accounts (R31-R34).
4. Calculate total liabilities (R35).
5. Set total assets (R28) to total liabilities.
6. Calculate current assets (R24).
7. Calculate cash (R21).
8. Calculate cash flow values (R51-R62).
9. Calculate actual cash change (RX3) to compare with the computed cash change (R62).

Additional Explanations

* The calculations reflect the following conventions:
    * all values are input as positive numbers,
    * income items on the income statement are positive and expense items are negative,
    * asset values are positive on the asset side of the balance sheet and liability values on the liability side are also positive,
    * sources of cash on the cash flow statement are positive and outflows negative.

* Working capital values are calculated as a function of sales. Accounts receivable are 48 days worth of sales. Inventories use the number 60, but since standard cost is 42% of sales the 60 is equivalent to 60/.42 = 143 days worth of sales.

* Accumulated depreciation (R27) is calculated by adding the 1980 opening balance (R27) and annual amounts for all years (R52) and accumulating the results.

* Accrued taxes (R32) are calculated as the accumulation of the opening 1980 balance (R32) plus the tax provision (R07) less tax payments (R60).

* Changes in asset accounts such as receivables, inventories, investments and fixed assets are calculated as prior year less current year values. If an asset increases then the difference will be negative which is consistent with the convention that cash usages are negative.

* Changes in liability accounps such as accounts payable and long term debt are calculated as current year less prior year values. If a liability increases then the difference will be positive, or a source of cash.

* The last three rules provide values for the last three rows which show the opening and closing cash position and the net change for each year.

```
                         PLAN80 EXAMPLE #3
            Projected Financial Statements and Cash Flow
                      (Thousands of Dollars)

                                      F1981    F1982    F1983    F1984
                                     ACTUAL     PLAN     PLAN     PLAN

CASH FLOW STATEMENT
===================
  Net Income                          298.0    312.1    336.0    360.0
  Depreciation                         70.0     90.0    120.0    140.0
  Change In Working Capital
    Accounts Receivable               -10.5     -7.9    -13.3    -13.3
    Inventories                       -13.2     -9.8    -16.7    -16.7
    Accounts Payable                    8.8      6.6     11.1     11.1
                                      -----    -----    -----    -----
      Total                           -14.9    -11.1    -18.9    -18.9
                                      -----    -----    -----    -----
  Change Due to
    Fixed Assets                     -130.0    -90.0   -130.0    -30.0
    Investments                           -   -120.0        -    -20.0
  Taxes
    ADD Provision                     130.9    143.9    166.0    188.0
    LESS Payments                    -100.0   -100.0   -100.0   -100.0
  Change in
    Long Term Debt                     50.0     50.0        -     50.0
                                      -----    -----    -----    -----
    Net Change                        303.9    274.9    373.1    569.1
                                      =====    =====    =====    =====


  CASH BALANCE
  ============
    Closing                           555.7    830.6   1203.7   1772.8
    Opening                           251.8    555.7    830.6   1203.7
                                      -----    -----    -----    -----
    Net Change                        303.9    274.9    373.1    569.1
                                      =====    =====    =====    =====
                         PLAN80 EXAMPLE #3
            Projected Financial Statements and Cash Flow
                       (Thousands of Dollars)

                             F1980   F1981   F1982   F1983   F1984
                            ACTUAL  ACTUAL    PLAN    PLAN    PLAN

INCOME STATMENT
===============
 Sales                       962.0  1041.0  1100.0  1200.0  1300.0
 Standard Cost              -404.0  -437.2  -462.0  -504.0  -546.0
                             -----   -----   -----   -----   -----
   Gross Margin              558.0   603.8   638.0   696.0   754.0
                             -----   -----   -----   -----   -----
 Advertising                 -77.0   -83.3   -88.0   -96.0  -104.0
 Administration              -88.5   -91.6   -94.0   -98.0  -102.0
                             -----   -----   -----   -----   -----
   Operating Profit          392.5   428.9   456.0   502.0   548.0
                             -----   -----   -----   -----   -----
 Tax Provision              -113.4  -130.9  -143.9  -166.0  -188.0
                             -----   -----   -----   -----   -----
   Net Income                279.1   298.0   312.1   336.0   360.0
                             =====   =====   =====   =====   =====

BALANCE SHEET - ASSETS
======================
 Cash                        251.8   555.7   830.6  1203.7  1772.8
 Accounts Receivable         128.3   138.8   146.7   160.0   173.3
 Inventories                 160.3   173.5   183.3   200.0   216.7
                             -----   -----   -----   -----   -----
   Current Assets            540.4   868.0  1160.6  1563.7  2162.8
                             -----   -----   -----   -----   -----
 Investments                 410.0   410.0   530.0   530.0   550.0
 Fixed Assets
   Gross                     720.0   850.0   940.0  1070.0  1100.0
   Accum Depreciation       -200.0  -270.0  -360.0  -480.0  -620.0
                             -----   -----   -----   -----   -----
   Total Assets             1470.4  1858.0  2270.6  2683.7  3192.8
                             =====   =====   =====   =====   =====

BALANCE SHEET - LIABILITIES
===========================
 Accounts Payable            106.9   115.7   122.2   133.3   144.4
 Accrued Taxes               413.4   444.3   488.1   554.1   642.1
 Lone Term Debt              250.0   300.0   350.0   350.0   400.0
 Equits                      700.1   998.1  1310.2  1646.3  2006.2
                             -----   -----   -----   -----   -----
   Total Liabilities        1470.4  1858.0  2270.6  2683.7  3192.8
                             =====   =====   =====   =====   =====
:TITLES
  1 "PLAN80 EXAMPLE #4"
  2 "Projected Financial Statements and Cash Flow"
  3 "(Thousands of Dollars)"
:COLUMNS
  F1980 (1) " F1980" "ACTUAL"
  F1981 (1) " F1981" "ACTUAL"
  F1982 (1) " F1982" "  PLAN"
  F1983 (1) " F1983" "  PLAN"
  F1984 (1) " F1984" "  PLAN"
:ROWS
  R01        "INCOME STATMENT"
             "==============="
             "  Sales"
  R02        "  Standard Cost"
  R03 (--)   "    Gross Margin"
  R04        "  Advertising"
  R05        "  Administration"
  R06 (--)   "    Operating Profit"
  R07        "  Tax Provision"
  R08 (-=)   "    Net Income"
  R21 (S)    "BALANCE SHEET - ASSETS"
             "======================"
             "  Cash"
  R22        "  Accounts Receivable"
  R23        "  Inventories"
  R24 (--)   "    Current Assets"
  R25        "  Investments"
  R26        "  Fixed Assets"
             "    Gross"
  R27        "    Accum Depreciation"
  R28 (-=)   "    Total Assets"
  R31 (S)    "BALANCE SHEET - LIABILITIES"
             "==========================="
             "  Accounts Payable"
  R32        "  Accrued Taxes"
  R33        "  Lone Term Debt"
  R34        "  Equits"
  R35 (-=)   "    Total Liabilities"
  R51 (P)    "CASH FLOW STATEMENT"
             "==================="
             "  Net Income"
  R52        "  Depreciation"
  R53        "  Change In Working Capital"
             "    Accounts Receivable"
  R54        "    Inventories"
  R55        "    Accounts Payable"
  R56 (--)   "      Total"
  R57        "  Change Due to"
             "    Fixed Assets"
  R58        "    Investments"
  R59        "  Taxes"
             "    ADD Provision"
  R60        "    LESS Payments"
  R61        "  Change in"
             "    Long Term Debt"
  R62 (-=)   "    Net Change"
  RX1 (SS)    "  CASH BALANCE"
              "  ============"
              "    Closing"
  RX2         "    Opening"
  RX3 (-=)    "    Net Change"
:DATA
  R01  =  962 1041 1100 1200 1300
  R25  =  410  410  530  530  550
  R26  =  720  850  940 1070 1100
  R27  =  200
  R32  =  300
  R33  =  250  300  350  350  400
  R34  =  421
  R52  =    0   70   90  120  140
  R60  =    0  100  100  100  100
:RULES
                                    : INCOME STATEMENT
  R02  =  -.42 * R01                  :standard cost
  R03  =   R01 + R02                  :gross margin
  R04  =  -.08 * R01                  :advertising
  R05  =   -50 - (.04 * R01)          :administration
  R06  =   R03 + R04 + R05            :operating profit
  R07  =    75 - (.48 * R06)          :tax provision
  R08  =   R06 + R07                  :net income
                                    : BALANCE SHEET
  R22  =   R01 * (48/360)             :accounts receivable
  R23  =   R01 * (60/360)             :inventories
  R27  =  -R27 - R52                  :depreciation
  R27  =  @CUM(R27)                   :accumulated depreciation
  R31  =   R01 * (40/360)             :accounts payable
  R32  =   R32 - R07 - R60            :accrued taxes
  R32  =  @CUM(R32)
  R34  =   R34 + R08                  :equity
  R34  =  @CUM(R34)
  R35  =  @SUM(R31..R34)              :total liabilities
  R28  =   R35                        :total assets
  R24  =   R28 - R26 - R27 - R25      :current assets
  R21  =   R24 - R23 - R22            :cash
:FOR COLUMNS=(F1981..F1984)
:RULES                              : CASH FLOW STATEMENT
  R51  =   R08                        :net income
  R53  =   R22(-1) - R22              :change in accounts receivable
  R54  =   R23(-1) - R23              :change in inventories
  R55  =   R31 - R31(-1)              :change in accounts payable
  R56  =  @SUM(R53..R55)              :change in working capital
  R57  =   R26(-1) - R26              :change in fixed assets
  R58  =   R25(-1) - R25              :change in investments
  R59  =  -R07                        :tax provision
  R60  =  -R60                        :tax payments
  R61  =   R33 - R33(-1)              :change in long term debt
  R62  =   R51 + R52 + @SUM(R56..R61) :net change in cash
  RX1  =   R21                        :closing cash balance
  RX2  =   R21(-1)                    :opening cash balance
  RX3  =   RX1 - RX2                  :net change in cash
:DISPLAY
```

EXAMPLE 4

Example 4 shows how to use PLAN80 for a problem in which the results of each period depend on those of the prior period. Investment income, for instance, depends on investment amount, which is a function of cash available at the end of the prior period. Similarly, interest expense is a function of debt, which is a function of prior period debt, prior period equity and current investment amount.

The example also shows how sum of digits depreciation is used to create cumulative depreciation and how the latter is used in computing current period depreciation.

Also included is a brief tax table which uses the @LOOKUP function.

Detailed Explanations

* The printed report shows how an editor can be used to make finishing touches on a report. All dashes, etc., have been removed from the Opening Balance column in the Income Statement section.

* The OPEN column has been used to store the original cash, debt and equity assumptions.

* New Physical Assets are acquired in 1981 and 1982. The Physical row contains the cumulative values of the new assets, and the CumDepr row shows cumulative depreciation.

* Current period depreciation is calculated for Deprec by subtracting prior from current period cummulative depreciation.

* The above calculations can be done one by one for all periods. The following calculations require that all calculations be done for each previous column before they are done for the current one. The FOR and INCLUDE statements are used to control the order of calculations.

* For each period investment is set to the prior period's investment plus any prior period cash in excess of 100.

* Debt is set to the maximum of prior period debt, prior period equity or 60% of the current period investment amount.

* Investment income and interest expense are calculated by multiplying appropriate rates and amounts.

* Taxes are calculated using a lookup table. The tax rate is 20% for income between 0 and 99, 30% for 99 to 199, etc.

```
                      PLAN80 EXAMPLE #4
                Reinvestment of Earnings Model

                          Opening
                          Balance    1981    1982    1983    1984    1985

Income Statement
  Investment Income                   270     271     295     353     429
  Interest Expense                    -81     -81     -90    -107    -129
  Admin Expense                      -175     -20     -22     -23     -25
  Depreciation                        -14     -26     -21     -17     -12
                                    -----   -----   -----   -----   -----
    Profit Before Tax                   -     144     162     206     263
  Taxes                                 -     -43     -49     -82    -105
                                    -----   -----   -----   -----   -----
    Net Income/Loss                     -     101     113     123     158
                                    =====   =====   =====   =====   =====

Balance Sheet - Assets
  Cash                       1000     104     179     293     353     414
  Investments                   -     900     904     983    1176    1429
  Physical Assets               -      50     100     100     100     100
  Accum Depreciation            -     -14     -40     -62     -79     -90
                            -----   -----   -----   -----   -----   -----
    Total Assets             1000    1040    1143    1314    1551    1852
                            =====   =====   =====   =====   =====   =====

Balance Sheet - Liabilities
  Debt                        500     540     542     600     714     857
  Equity                      500     500     600     714     837     995
                            -----   -----   -----   -----   -----   -----
    Total Liabilities        1000    1040    1143    1314    1551    1852
                            =====   =====   =====   =====   =====   =====

Assumptions
  New Physical Assets           -      50      50       -       -       -
  Yield on Investment           -   0.300   0.300   0.300   0.300   0.300
  Cost for Debt                 -   0.150   0.150   0.150   0.150   0.150
  Tax Bracket #1                -       -       -       -       -       -
  Tax Bracket #2                -      99      99      99      99      99
  Tax Bracket #3                -     199     199     199     199     199
  Tax Rate #1                   -   0.200   0.200   0.200   0.200   0.200
  Tax Rate #2                   -   0.300   0.300   0.300   0.300   0.300
  Tax Rate #3                   -   0.400   0.400   0.400   0.400   0.400
```

```
:OPTIONS
  ROWWID(21)
:TITLES
  1 "PLAN80 EXAMPLE #4"
  2 "Reinvestment of Earnings Model"
:COLUMNS
  OPEN           "Opening" "Balance"
  Y1981                " "    "1981"
  Y1982                " "    "1982"
  Y1983                " "    "1983"
  Y1984                " "    "1984"
  Y1985                " "    "1985"
:ROWS
  InvInc         "Income Statement"
                 "  Investment Income"
  IntExp         "  Interest Expense"
  Admin          "  Admin Expense"
  Deprec         "  Depreciation"
  PBT      (-)   "    Profit Before Tax"
  Taxes          "  Taxes"
  NET      (-=)  "    Net Income/Loss"
  Cash     (s)   "Balance Sheet - Assets"
                 "  Cash"
  Invest         "  Investments"
  Physical       "  Physical Assets"
  CumDepr        "  Accum Depreciation"
  ASSETS   (-=)  "    Total Assets"
  Debt     (s)   "Balance Sheet - Liabilities"
                 "  Debt"
  Equity         "  Equity"
  LIABIL   (-=)  "    Total Liabilities"
  NewPhys  (s)   "Assumptions"
                 "  New Physical Assets"
  Yield    (3)   "  Yield on Investments"
  IntRate  (3)   "  Cost for Debt"
  Bracket1       "  Tax Bracket #1"
  Bracket2       "  Tax Bracket #2"
  Bracket3       "  Tax Bracket #3"
  TRate1   (3)   "  Tax Rate #1"
  TRate2   (3)   "  Tax Rate #2"
  TRate3   (3)   "  Tax Rate #3"
:DATA
  Admin     =     *  -175 -20 (*108%)
  Cash      =  1000
  NewPhys   =     *    50  50
  Debt      =   500
  Equity    =   500
  Yield     =     *    30% +
  IntRate   =     *    15% +
  Bracket1  =     *     0  +
  Bracket2  =     *    99  +
  Bracket3  =     *   199  +
  TRate1    =     *    20% +
  TRate2    =     *    30% +
  TRate3    =     *    40% +
:RULES
  Physical      = @CUM(NewPhys)
  CumDepr       = -@SOD(NewPhys,6)

:FOR COLUMNS=(Y1981)
:INCLUDE "PROC04.TXT"  <--------
:FOR COLUMNS=(Y1982)           |
:INCLUDE "PROC04.TXT"  <--------|
:FOR COLUMNS=(Y1983)           |
:INCLUDE "PROC04.TXT"  <--------|
:FOR COLUMNS=(Y1984)           |
:INCLUDE "PROC04.TXT"  <--------|
:FOR COLUMNS=(Y1985)           |
:INCLUDE "PROC04.TXT"  <--------|
                               |
:DISPLAY                       |
                               |
                               |
                               |
     ------------------------------------------------------------
     | * * *  FILE:  PROC04.TXT  * * *                          |
     |                                                          |
     | :RULES                                                   |
     |   Deprec  =  CumDepr - CumDepr(-1)                       |
     |   Invest  =  Invest(-1)+@MAX(100,Cash(-1))-100           |
     |   Debt    =  @MAX(Debt(-1),Equity(-1),(60%*Invest))      |
     |   InvInc  =  Yield * Invest                              |
     |   IntExp  =  -IntRate * Debt                             |
     |   PBT     =  Admin + InvInc + Deprec + IntExp            |
     |   Taxes   =  -PBT * @LOOKUP(PBT,Bracket1..Bracket3,TRate1) |
     |   NET     =  PBT + Taxes                                 |
     |   Equity  =  Equity(-1) + NET                            |
     |   LIABIL  =  Debt + Equity                               |
     |   ASSETS  =  LIABIL                                      |
     |   Cash    =  ASSETS - CumDepr - Physical - Invest        |
     ------------------------------------------------------------
```

EXAMPLE 5

Example 5 is an interactive application which allows you to input investment by period in one column and income by period in another. A third column shows the discounted value of the investment and income amounts by period. Subtotals show the total of the investment and income values and confirm that the sum of the discounted amounts is zero. The discount factor used to achieve this zero total is called the internal rate of return.

```
                             PLAN80 EXAMPLE #5
                       Internal Rate of Return Model

                   P01   P02   P03   P04   P05   P06   P07   P08   P09   P10


Investment        -400     -     -     -     -     -     -     -     -     -
Income               -   100   110   121   133   146   161   177   195   214
Cash Flow         -400   100   110   121   133   146   161   177   195   214
Cumulative        -400  -300  -190   -69    64   211   372   549   744   958
Internal Rate
 of Return (%)       -     -     -     -   5.9  14.9  20.7  24.5  27.1  29.0
```

```
:TITLES
  1 "PLAN80 EXAMPLE #5"
  2 "Internal Rate of Return Model"
:COLUMNS
  P01
  P02
  P03
  P04
  P05
  P06
  P07
  P08
  P09
  P10
  P11
  P12
:ROWS
  INVEST         "Investment"
  INCOME         "Income"
  CASHFLOW       "Cash Flow"
  CUMCF          "Cumulative"
  IRRATE     (1) "Internal Rate"
                 " of Return (%)"
:OPTIONS
  ROWWID(14)
  COLWID(5)
:DATA
  INVEST   = -400
  INCOME   =    0  100 (*1.10)
:INTERACTIVE
:RULES
  CASHFLOW =  INCOME+INVEST
  CUMCF    =  @CUM(CASHFLOW)
  IRRATE   =  @IRR(CASHFLOW)
:DISPLAY
```

EXAMPLE 6

Example 6 is an administrative cost center budgeting example. The @SUM function is an extremely convenient shorthand technique for summing large numbers of rows or columns. Note also how occupancy expense is calculated as a function of headcount. Similar techniques could be used to build flexible budgets or to allocate overhead.

```
                     PLAN80 EXAMPLE #6
            ADMINISTRATIVE BUDGET FOR COST CENTER 1001

                       1ST      2ND      3RD      4TH      FULL
                       QTR      QTR      QTR      QTR      YEAR

HEADCOUNT
 EXEMPT                2.0      2.0      2.0      3.0       9.0
 NONEXEMPT             1.0      1.0      1.0      1.0       4.0
                     -----    -----    -----    -----     -----
  TOTAL                3.0      3.0      3.0      4.0      13.0

EXPENSE
 SALARIES             24.0     25.2     25.2     31.8     106.2
 FRINGES               7.2      7.6      7.6      9.5      31.9
 OVERTIME              2.0      3.0      2.0      2.3       9.3
 SUPPLIES              0.7      0.8      0.8      0.8       3.1
 RENTALS               3.5      3.5      3.5      3.5      14.0
 OCCUPANCY             6.1      6.1      6.1      6.9      25.2
 TELEPHONE             2.8      2.8      2.8      2.8      11.2
 TRANSFERS             3.6      4.1      3.4      3.5      14.6
                     -----    -----    -----    -----     -----
  TOTAL               49.9     53.1     51.4     61.1     215.5
                     =====    =====    =====    =====     =====
:TITLES
  1 "PLAN80 EXAMPLE #6"
  2 "ADMINISTRATIVE BUDGET FOR COST CENTER 1001"
:COLUMNS
  Q1   (1)  "1ST"  "QTR"
  Q2   (1)  "2ND"  "QTR"
  Q3   (1)  "3RD"  "QTR"
  Q4   (1)  "4TH"  "QTR"
  YEAR (1)  "FULL" "YEAR"
:ROWS
  H01       "HEADCOUNT"
            " EXEMPT"
  H02       " NONEXEMPT"
  H99  (-)  "  TOTAL"
  S01  (S)  "EXPENSE"
            " SALARIES"
  S02       " FRINGES"
  S03       " OVERTIME"
  S04       " SUPPLIES"
  S05       " RENTALS"
  S06       " OCCUPANCY"
  S07       " TELEPHONE"
  S08       " TRANSFERS"
  S99  (-=) "  TOTAL"
:DATA
  H01  =     2    2    2    3
  H02  =     1    1    1    1
  S01  =  24.0 25.2 25.2 31.8
  S03  =   2.0  3.0  2.0  2.3
  S04  =    .7   .8   .8   .8
  S05  =   3.5  3.5  3.5  3.5
  S07  =   2.8  2.8  2.8  2.8
  S08  =   3.6  4.1  3.4  3.5
:RULES
  H99  =  H01 + H02
  S02  =  .30 * S01
  S06  =  4.0 + (.8*H01) + (.5*H02)
  S99  =  @SUM(S01..S08)
  YEAR =  @SUM(Q1..Q4)
:DISPLAY
```

EXAMPLE 7

Example 7 shows the consolidation of three cost center budgets into a summary. It should be compared to Example 6, which has one DATA and one RULES section. Example 7, in contrast, uses the INCLUDE statement to read title line #3 and a DATA section for each cost center. Note that for each cost center we

- initialize data values to zero,
- include the title and data for the particular cost center,
- include a common set of rules,
- display the results, and
- put the results into a file.

Then for the consolidation we

- initialize data values to zero,
- get each previously saved file using the "add" option,
- introduce an appropriate title, and
- display the results.

The page containing PLAN80 statements is supplemented with diagrams to help you understand the consolidation.

```
                         PLAN80 EXAMPLE #7
                       BUDGET CONSOLIDATION
                SUMMARY OF COST CENTERS 1001-1003

                          1ST      2ND      3RD      4TH     FULL
                          QTR      QTR      QTR      QTR     YEAR

HEADCOUNT
 EXEMPT                  10.0     10.0     10.0     13.0     43.0
 NONEXEMPT               24.0     25.0     25.0     26.0    100.0
                        -----    -----    -----    -----    -----
  TOTAL                  34.0     35.0     35.0     39.0    143.0

EXPENSE
 SALARIES               162.0    165.7    168.2    182.4    678.3
 FRINGES                 48.6     49.7     50.5     54.7    203.5
 OVERTIME                14.3     14.9     12.8     14.7     56.7
 SUPPLIES                12.1     11.4     12.3     11.8     47.6
 RENTALS                 23.5     23.5     23.5     23.5     94.0
 OCCUPANCY               32.0     32.5     32.5     35.4    132.4
 TELEPHONE               16.4     16.4     16.4     16.4     65.6
 TRANSFERS                  -        -        -        -        -
                        -----    -----    -----    -----    -----
  TOTAL                 308.9    314.1    316.2    338.9   1278.1
                        =====    =====    =====    =====    =====
                 ----------------------------------------------------------
                |                    PLAN80 EXAMPLE #7                      |
                |                   BUDGET CONSOLIDATION                    |
                |                     COST CENTER 1003                      |
                |                                                           |
                |                      1ST     2ND     3RD     4TH    FULL  |
                |                      QTR     QTR     QTR     QTR    YEAR  |
                |                                                           |
                | HEADCOUNT                                                 |
                |  EXEMPT              2.0     2.0     2.0     3.0     9.0  |
                |  NONEXEMPT          21.0    22.0    22.0    23.0    88.0  |
                |                    -----   -----   -----   -----   -----  |
                |   TOTAL             23.0    24.0    24.0    26.0    97.0  |
                |                                                           |
         -----------------------------------------------------------        |
        |                    PLAN80 EXAMPLE #7                     | 306.3  |
        |                   BUDGET CONSOLIDATION                   |  91.9  |
        |                     COST CENTER 1002                     |  26.1  |
        |                                                          |  37.2  |
        |                      1ST     2ND     3RD     4TH    FULL |  50.0  |
        |                      QTR     QTR     QTR     QTR    YEAR |  67.2  |
        |                                                          |  31.2  |
        | HEADCOUNT                                                | -47.2  |
        |  EXEMPT              6.0     6.0     6.0     7.0    25.0 | -----  |
        |  NONEXEMPT           2.0     2.0     2.0     2.0     8.0 | 562.7  |
        |                    -----   -----   -----   -----   ----  | =====  |
        |   TOTAL              8.0     8.0     8.0     9.0    33.0 | ------
        |                                                          |
 -----------------------------------------------------------       |
|                    PLAN80 EXAMPLE #7                     | 265.8 |
|                   BUDGET CONSOLIDATION                   |  79.7 |
|                     COST CENTER 1001                     |  21.3 |
|                                                          |   7.3 |
|                      1ST     2ND     3RD     4TH    FULL |  30.0 |
|                      QTR     QTR     QTR     QTR    YEAR |  40.0 |
|                                                          |  23.2 |
| HEADCOUNT                                                |  32.6 |
|  EXEMPT              2.0     2.0     2.0     3.0     9.0 | ----- |
|  NONEXEMPT           1.0     1.0     1.0     1.0     4.0 | 499.9 |
|                    -----   -----   -----   -----   ----- | ===== |
|   TOTAL              3.0     3.0     3.0     4.0    13.0 | ------
|                                                          |
| EXPENSE                                                  |
|  SALARIES           24.0    25.2    25.2    31.8   106.2 |
|  FRINGES             7.2     7.6     7.6     9.5    31.9 |
|  OVERTIME            2.0     3.0     2.0     2.3     9.3 |
|  SUPPLIES            0.7     0.8     0.8     0.8     3.1 |
|  RENTALS             3.5     3.5     3.5     3.5    14.0 |
|  OCCUPANCY           6.1     6.1     6.1     6.9    25.2 |
|  TELEPHONE           2.8     2.8     2.8     2.8    11.2 |
|  TRANSFERS           3.6     4.1     3.4     3.5    14.6 |
|                    -----   -----   -----   -----   ----- |
|   TOTAL             49.9    53.1    51.4    61.1   215.5 |
|                    =====   =====   =====   =====   ===== |
 -----------------------------------------------------------
```

```
                                     ------------------------------------
:TITLES                              | :TITLES                          |
  1 "PLAN80 EXAMPLE #7"              |   3  "COST CENTER 1001"          |
  2 "BUDGET CONSOLIDATION"   ------  | :DATA                            |
:COLUMNS                    |        |   H01     =    2    2    2    2  |
  Q1   (1)  "1ST"  "QTR"    |        |   H02     =    1    1    1    1  |
  Q2   (1)  "2ND"  "QTR"    |        |   S01     = 24.0 25.2 25.2 31.8  |
  Q3   (1)  "3RD"  "QTR"    |        |   S03     =  2.0  3.0  2.0  2.3  |
  Q4   (1)  "4TH"  "QTR"    |        |   S04     =   .7   .8   .8   .8  |
  YEAR (1)  "FULL" "YEAR"   |        |   S05     =  3.5  3.5  3.5  3.5  |
:ROWS                       |        |   S07     =  2.8  2.8  2.8  2.8  |
  H01       "HEADCOUNT"     |        |   S08     =  3.6  4.1  3.4  3.5  |
            "EXEMPT"        |        ------------------------------------
  H02       " NONEXEMPT"    |
  H99  (-)  "  TOTAL"       |
  S01  (S)  "EXPENSE"       |        ------------------------------------
            " SALARIES"     |        | :TITLES                          |
  S02       " FRINGES"      |        |   3  "COST CENTER 1002"          |
  S03       " OVERTIME"     |    --  | :DATA                            |
  S04       " SUPPLIES"     |   |    |   H01     =    6    6    6    7  |
  S05       " RENTALS"      |   |    |   H02     =    2    2    2    2  |
  S06       " OCCUPANCY"    |   |    |   S01     = 64.0 65.0 66.0 70.8  |
  S07       " TELEPHONE"    |   |    |   S03     =  5.0  5.0  5.0  6.3  |
  S08       " TRANSFERS"    |   |    |   S04     =  1.7  1.8  2.3  1.5  |
  S99  (-=) "  TOTAL"       |   |    |   S05     =  7.5  7.5  7.5  7.5  |
:INITIALIZE                 |   |    |   S07     =  5.8  5.8  5.8  5.8  |
:INCLUDE "CC1001.TXT" <----     |    |   S08     =  7.6  8.1  8.4  8.5  |
:INCLUDE "PROC07.TXT" <--       |    ------------------------------------
:PUT "T1001"              |     |
:INITIALIZE               |     |
:INCLUDE "CC1002.TXT" <--------      ------------------------------------
:INCLUDE "PROC07.TXT" <--|           | :TITLES                          |
:PUT "T1002"              |          |   3  "COST CENTER 1003"          |
:INITIALIZE               |          | :DATA                            |
:INCLUDE "CC1003.TXT" <-----------   |   H01     =    2    2    2    2  |
:INCLUDE "PROC07.TXT" <--|           |   H02     =   21   22   22   23  |
:PUT "T1003"              |          |   S01     = 74.0 75.5 77.0 79.8  |
:                         |          |   S03     =  7.3  6.9  5.8  6.1  |
: * *  NOW CONSOLIDATE  * *          |   S04     =  9.7  8.8  9.2  9.5  |
:                         |          |   S05     = 12.5 12.5 12.5 12.5  |
:INITIALIZE               |          |   S07     =  7.8  7.8  7.8  7.8  |
:GET "T1001" ADD          |          |   S08 = -11.2 -12.2 -11.8 -12.8  |
:GET "T1002" ADD          |          ------------------------------------
:GET "T1003" ADD          |
:TITLES                   |
  3 "SUMMARY OF COST CENTERS 1001-1003"
:DISPLAY                  |       ----------------------------------------
                          |      | :RULES                                 |
                          -----  |   H99   = H01 + H02                    |
                                 |   S02   = S01 * .30                    |
                                 |   S06   = 4.0 + (.8*H01) + (.5*H02)    |
                                 |   S99   = @SUM(S01..S08)               |
                                 |   YEAR = @SUM(Q1..Q4)                  |
                                 | :DISPLAY                               |
                                  ----------------------------------------
```

APPENDIX B

PLAN80 INSTALLATION

WHAT YOU WILL NEED

To operate PLAN80 you will need

* a CP/M based computer with at least 56K of memory,
* at least two disk drives,
* a terminal with cursor addressing and the clear screen function,
* a disk containing the CP/M system,
* the PLAN80 disk containing the PLAN80 system and examples,
* at least two blank disks (preferably several),
* an editor (which creates and modifies text files), and
* this manual.

Equipment such as the following would be welcome additions to your system and should be fully compatible with PLAN80.

DISK DRIVES

One drive is adequate to operate most PLAN80 applications, but you will need a second for copying files. In some instances a hard disk might be useful. The capacity of the smallest hard disk is about 20 times that of an 8" single density floppy disk.

PRINTERS

PLAN80 may be operated without a printer, but you will want to obtain one in the near future if you don't already own one. Your dealer can show you a range of printers offering good to superior print quality and slow to high print speeds.

WHAT YOU MUST DO

You need to perform several steps before operating PLAN80.

1. Format two or more blank disks. One will become your PLAN80 master and the others will become operating copies of the system.
2. Place a copy of the CP/M operating system on each newly formatted disk.
3. Copy certain CP/M and PLAN80 files on one of the newly formatted disks. That disk will become your PLAN80 master.
4. Establish a terminal control file on the PLAN80 master.
5. Make operating copies of the new PLAN80 master.

Each step should take only a few minutes and is described in detail on the following pages.

STEP 1 - FORMAT BLANK DISKS

You should have enough blank disks to create a PLAN80 master plus at least one operating copy for each major application area (e.g. budgeting, sales forecasting, tax planning, etc.).

Before your computer can use a new disk it must receive marks on its surface which indicate where information is to be stored. The process of placing these marks is called formatting.

Formatting is done by a program on your CP/M source disk whose name is probably FORMAT.COM.

You should review your computer's manuals for exact instructions on how to format blank disks if you are not already familiar with this step. Don't forget to place a tab over the write-protect notch of each disk if your system uses the normal scheme for protecting 8" disks.

STEP 2 - COPY THE CP/M SYSTEM

You should use a program on your CP/M source disk, frequently called SYSGEN, to place a copy of the CP/M system on each disk formatted in the previous step.

You should review your computer manuals for exact instructions on how to place the CP/M operating system on disks if you are not already familiar with this step.

STEP 3 - CREATE A PLAN80 MASTER

In this step you will copy files onto one of your newly formatted disks which contains the CP/M operating system. That disk will become the PLAN80 master from which you will create operating copies. Place a newly formatted disk which contains the CP/M operating system in drive A.

Now place a CP/M disk with the file PIP.COM in drive B. You will use the PIP program to place a copy of itself on the disk in drive A. Enter the command

```
  A>B:PIP A:=B:PIP.COM
```

Now you will use the new copy of PIP.COM to copy other files onto the PLAN80 master. You should copy the CP/M program STAT.COM

```
  A>PIP A:=B:STAT.COM
```

If you have a directory listing program which you prefer to the CP/M DIR command you may want to copy it also.

Then you should copy the one or more files which comprise your most capable editor. You will create input for PLAN80 just as if you were writing a letter.

Lastly you will need to copy files from the original PLAN80 disk. Place the original PLAN80 disk in drive B and use the directory command

```
  A>DIR B:
```

to view its contents.

You should copy the PLAN80 system files, using the command

```
  A>PIP PLAN80.*=B:PLAN80.*
```

You should then copy the PLAN80 examples using the command

```
  A>PIP A:=B:*.TXT
```

Lastly you should look at the file TERMINAL.LST using either your editor or the CP/M command

```
  A>TYPE B:TERMINAL.LST
```

TERMINAL.LST contains the name of each terminal for which a terminal control file exists and indicates the values used for each terminal control function. If you find a file for your terminal then copy it with the command

```
  A>PIP CONTROL.TRM=B:xxxxxx.TRM
```

where xxxxxx indicates your particular terminal.

If your terminal is not named then you should copy the INSTALL program with the command

```
  A>PIP A:=B:INSTALL.*
```

and use it to create and test a terminal control file as outlined in step 4.

Contents of Original PLAN80 Disk

File Name	Description
Category I - PLAN80 System	
PLAN80.COM	
PLAN80.OV1	
PLAN80.OV2	
Category II - PLAN80 Examples	
EX01.TXT	Five year Profit Projection
EX02.TXT	Projected Sales and Gross Margin
EX03.TXT	Projected Financial Statements and Cash Flow
EX04.TXT	Reinvestment of Earnings Model
EX05.TXT	Internal Rate of Return
EX06.TXT	Administrative Cost Center Budget
EX07.TXT	Budget Consolidation
Category III - Terminal Control Files	
TERMINAL.LST	List of Terminal Control Files Available
INSTALL.COM	Creates Terminal Control Files
ADDS.TRM	ADDS Regent
ADM3A.TRM	Lear Siegler ADM-3A
CROMEMCO.TRM	Cromemco 3102
HAZ1500.TRM	Hazeltine 1500
IBM.TRM	IBM 3101
MICRO.TRM	Microterm ACT IV/V
SOROC.TRM	Soroc IQ120
SBRAIN.TRM	Intertec Superbrain
TELEVID.TRM	Televideo 912/920
VECTOR.TRM	Vector Flashwriter II
ZENITH.TRM	Zenith WH19/Heath H19

STEP 4 - ESTABLISH A TERMINAL CONTROL FILE

If you copied one of the terminal control files then you may skip this step. If you did not find your terminal listed then you will use the INSTALL program to create, test and/or change the file CONTROL.TRM.

Use of the INSTALL program is described in detail on the following pages. You should read through the section and then find the information required in the user manual for your terminal or contact your local computer dealer.

When you have created and tested the file CONTROL.TRM you should delete the INSTALL program with the command

```
A>ERA INSTALL.*
```

(Note: If you have a terminal connected to a special video board in your computer you may need to contact your PLAN80 distributor for alternative installation instructions.)

You should now look at the directory of your new PLAN80 master and insure that it contains

PIP.COM
STAT.COM
an editor
the PLAN80 system
the PLAN80 examples
CONTROL.TRM

Your PLAN80 master is complete. It is a good idea to write-protect the master before making copies. This usually entails removing a tab covering the write-protect notch on 8" disks or adding a tab to smaller disks.

STEP 5 - MAKE OPERATING COPIES

To make an operating copy place the PLAN80 master in drive A and a formatted disk containing the CP/M system in drive B and enter

```
A>PIP B:=*.*
```

Repeat the copying process for each disk formatted in step 1. You should now be ready to operate PLAN80.

OPERATING THE INSTALL PROGRAM

The INSTALL program creates or modifies a file which tells PLAN80 how to move your terminal's cursor, to clear the screen and to clear to the end of a line. Before using the program you should understand in general terms how terminals are controlled.

Your computer sends units of information called bytes to your terminal. Bytes always have a value between zero and 255. If the value is between 32 and 125 then a character is printed at the current cursor position and the cursor automatically moves forward one position. A value of 32 corresponds to a blank, 33 to an exclamation mark, 89 to a upper case Y, etc., in accordance with an industry standard called the ASCII Character Codes.

Bytes with values of less than 32 are used for control purposes, but nearly every terminal manufacturer uses different codes for each control function. For instance, a byte with value 12 will clear the screen of some terminals while others may require a sequence of seven bytes to accomplish this task.

The INSTALL program will help you specify and test sequences for the following control functions:

* position the cursor,
* clear the screen,
* clear to the end of a line,
* clear to the end of the screen, and
* turn highlighting on and off.

Your terminal need not have all of the above functions, but it must have the first two. (Almost all terminals have these two functions.) The next two functions can be simulated by writing blanks. The last function is available on most terminals as either highlighting or as inverse printing (e.g. dark characters on a light background vs. light characters on a dark background). Highlighting of either type is desirable for most PLAN80 displays, but not required. The graphics analysis mode will require a sharp eye, however, if you do not have highlighting.

Terminals are usually controlled by two byte sequences (plus row and column data for cursor positioning). The first byte indicates that a control sequence has begun, and the second indicates the particular control function desired.

Cursor positioning has a number of complexities. First, the order in which row and column information is expected is not standard. Second, most terminals expect a number, usually 32, to be added to the actual column and row number. Lastly, the row and column data may be separated and/or terminated by other bytes. To handle these complexities INSTALL treats the cursor positioning in five parts:

* an opening sequence,
* the row (or column) number plus an offset value,
* a possible second sequence,
* the column (or row) number plus an offset value, and
* a possible ending sequence.

Each sequence may be up to nine bytes long, and an empty sequence is used when a function is not supported or if the sequence is *not needed (as in the ending sequence for cursor positioning, for* example).

You operate the INSTALL program by entering the command

```
A>INSTALL
```

In a few seconds the screen will be cleared and you will see a menu from which you may select a field to change

```
TERMINAL INSTALLATION (C) 1981
BUSINESS PLANNING SYSTEMS INC

TERMINAL DESCRIPTION:
A - Name                    SOROC 120
CURSOR POSITIONING
B - Leadin Sequence         27 61
C - Row (or Col) + Offset   ROW+32
D - Second Sequence
E - Col (or Row) + Offset   COL+32
F - Ending Sequence
OTHER FUNCTIONS:
G - Clear Screen and Home   27 42 + 8 nulls
H - Clear to End of Line    27 84
I - Clear to End of Screen  27 89
J - Highlighting On         27 40
K - Highlighting Off        27 41
MISCELLANEOUS INFO:
L - Number of Rows          24
M - Number of Columns       80
N - Cursor Origin           0
O - Use ASCII Characters    NO

PRESS SPACEBAR OR INPUT CODE -->
```

If you press the spacebar you will see a continuation as follows:

```
T - TEST ABOVE SEQUENCES
R - READ CURRENT FILE
W - WRITE NEW FILE
X - EXIT

PRESS SPACEBAR OR INPUT CODE -->
```

If you press the spacebar you will see the first part of the menu again.

CHANGING TERMINAL CONTROL VALUES

You may press any of the keys A through O to change the values shown on the TERMINAL INSTALLATION menu. Furthermore you may change values in any order desired, and you may choose to enter or exit the test mode at any time.

After you enter any new value the first part of the menu will be redisplayed.

ENTERING A TERMINAL NAME

If you press A to input a new name you will be asked to

INPUT DESCRIPTION -->

You may enter up to 33 characters and may back up a space to remove any characters typed incorrectly. Press RETURN when you are satisfied with what has been entered.

CHANGING SEQUENCES

If you press any of the keys B,D,F or G through K to input a new sequence you will be prompted to

```
  INPUT SEQUENCE -->
```

and then may enter up to nine values as in

```
  27 89
```

or

```
  27 91 72 27 91 50 74
```

Values may be entered as numbers as shown above or as characters. For example, if your terminal's manual indicates the sequence ESC Y you may press the ESCape and upper case Y keys instead of numbers. (Note: Usually you will need to input upper case letters.) When the sequence is redisplayed (after you press RETURN) you will see the characters converted to values (27 89 in the case of ESC Y).

Each number value specified should be less then 256 and greater than zero. (A zero value would terminate the sequence.)

Some terminals require a brief delay at the end of one function (such as clearing the screen) before beginning another. This is usually done by writing a number of bytes of zero value (called nulls) at the end of the first sequence. If your terminal needs nulls you can add up to 255 of them at the end of a sequence by entering a plus sign and the number of nulls as in

```
  27 42 + 10
```

Press the RETURN key when you are satisfied with what you have entered. You may backup a space to delete any character(s) entered incorrectly before you press the the RETURN key.

CHANGING ROW OR COLUMN ORDER AND OFFSET

If you press the C key you may change the order in which the row and column number is provided for cursor positioning and the offset which will be added to the row or column number. You will be asked to

```
  INPUT "R" OR "C" -->
```

Press R if row is required first. Otherwise press C. Most terminals require row information first. Then you will be asked to

```
  INPUT OFFSET AMOUNT -->
```

Generally the amount will be 32, but some terminals use zero, and a few use 128 or some other value. Enter the value and press RETURN when you are done.

If you pressed the E key initially you will see the same two questions.

When you specify row order for either of the two lines then the other line is automatically set to column order, and vice versa.

SETTING SCREEN SIZE

If you press L or M to specify the number of rows or columns you will be asked to

```
  INPUT NUMBER -->
```

You should enter the appropriate number and press RETURN.

SETTING CURSOR ORIGIN

Most terminals refer to the upper left corner as row zero and column zero. There are a few exceptions, however. If you press N you will be asked to

```
  INPUT NUMBER -->
```

You should enter the appropriate number and press RETURN.

SETTING THE ASCII CHARACTER OPTION

If you press O you will be asked if your terminal expects your computer to

```
  SEND ASCII CHARACTERS?  -->
```

to indicate cursor positioning information. Most terminals expect this information in the form of a number, but a few expect ASCII characters.

TEST ABOVE SEQUENCES

You may press T to test the effect of existing sequences and values. There are five tests, one for each of the following functions, and the tests occur in the order shown

- cursor addressing,
- clear screen and home,
- clear to end of line,
- clear to end of screen, and
- highlighting.

After each test you will be asked

```
  WAS TEST SUCCESSFUL? (Y/N)
```

If you respond Y you will advance to the next test. Otherwise the main menu will appear in order that you may change values and retest.

READ CURRENT FILE

If you press R the INSTALL procram will get data from the file CONTROL.TRM if it exists. Data items in that file will replace all data items currently shown on the TERMINAL CONTROL menu.

WRITE NEW FILE

If you press W the INSTALL program will create a file CONTROL.TRM. Any existing file named CONTROL.TRM will be deleted.

EXIT

Press X to leave the INSTALL program. Note that a new CONTROL.TRM file is not automatically created. You must press W, as noted above, to write a new file.

APPENDIX C

OPERATING PLAN80

USE OF THE WORDS "ENTER" AND "KEY"

The word "enter", as used in this manual, means to strike a series of keys. At any point you may use the left arrow or backspace key to delete one or more characters typed in error. Then you should press RETURN when you are done.

The word "key" means that you should press the key(s) indicated.

OPERATING PLAN80

Before operating PLAN80 you must have an operating copy of the PLAN80 system prepared as discussed in Appendix B.

To run a PLAN80 application turn on your computer, insert an operating copy of the PLAN80 system, and, if necessary, press RESET. You should see the CP/M prompt

```
  A>
```

Enter PLAN80. After a few seconds your screen will be cleared and you will see the PLAN80 prompt

```
  PLAN80   VERSION 2.3   COPYRIGHT (C) 1982
      BUSINESS PLANNING SYSTEMS, INC.

  INPUT NAME OF PROCEDURE FILE:
    --->
```

EXECUTING EXAMPLE 2

Enter EX02. Each statement appears as it is being processed. You may stop processing by holding down the CTRL key while you key S. This sequence is called CTRL-S and operates as a switch turning processing on and off. Go ahead and try it a few times.

DATA DISPLAY MODE

After you see the DISPLAY statement you will be in the data display mode and will see the Example 2 results in tabular form.

CURSOR MOVEMENT

Key lower case k twice. Note that the cursor moves right one column each time that k is pressed. Now press 3k. Note that the cursor moves 3 positions at once. Key 4m and the cursor will move down by four positions. Key 3j and 2i and the cursor will move left three columns and up two rows.

Note the exact row/column location of the cursor and press p. The cursor will remain at the same row/column location, but the screen will be displayed placing the cursor at the upper left corner.

Move the cursor right and down several columns and rows. Key h and the cursor will return to the upper left corner. Key h again and the screen will be redisplayed starting with the first row/column location.

Move the cursor right and down again several rows and columns. Key upper case K, and the cursor will move to the far right of the screen. Key J and it will move to the far left. Similarly upper case I and M will move the cursor to the top and bottom of the screen.

You may wish to experiment for a few minutes with the above keys. You will also find that pressing one of the upper case I,J,K or M keys when the cursor is already at the edge of the screen will move the cursor to the first or last row or column of your model.

You will also discover combinations of keystrokes that make it possible to scroll in any direction. You might want to try Mp for down, 16i for up, Kp for right and 6j for left.

DISPLAY OPTIONS MODE

Press the ESCape key to enter the display options mode.

Press the spacebar to see a list of available options. Many options affect the way reports appear when displayed or printed. Others permit you to enter the graphics analysis mode, create DATA files and exit the display mode or return to the CP/M system. These options are covered in detail in Chapter 6.

GRAPHICS ANALYSIS MODE

Key g to enter the graphics analysis mode. Key d and you will see a graph of the first row.

Key >>> (upper or lower case) to bring the highlighted line on the right down to the item marked row2. Press the spacebar to select this item. Now key >> to select the second row and press the spacebar to leave the item.

Key > to move down another line. Press the spacebar, key 3> to select the third row, and key D for a graph of all three rows.

Key 3< to move up to the type item. Press the spacebar and the type should change from scatter to bar #1. Key d to display a bar graph of type 1. Press the spacebar again and the type will change to bar #2. Key d to display a bar graph of type 2. Press the spacebar again and the type will change to cumulative. Again key d.

Press ESCape to return to the display mode.

Note: In Version 2.0 you cannot display graphs on your printer.

DATA INPUT MODE

From the display mode you may press the spacebar to enter the data input mode and then may input new values in any row or column starting from the current cursor position. Press the spacebar and note the message at the bottom of the screen

```
  Input Value:
```

Key 333 and the spacebar to enter 333 without moving the cursor position. Note that you have left the data input mode. Press the spacebar again to reenter.

Key 444k to enter 444 and to move the cursor one position to the right. Note that you are still in the data input mode. Key 555k and 666k to enter two more values. Now key rk. Whenever you use the r key it recalls the last value entered.

Key mJ to move to the start of the next row and then key 333:4k. The 4 following the colon causes the number 333 to be entered four times.

Key mJ again and then key 333*108%, and 360 will be entered. You may enter any calculation involving two numbers separated by +,-,* or /.

Key mJ again and then key 333*108%:4k. You will see four numbers entered each of which is 8% greater than the previous number.

You may use features covered in the above examples to enter groups of values in any direction.

You may press the spacebar at any time to leave the data input mode.

INTERRUPTING PLAN80

Key p and then key m a few times after only a few lines have been displayed. You should be able to interrupt the display and scroll rapidly to any point. (Note: some terminals or computers may not give you this feature.) You may use any cursor control keys to interrupt the display.

You can generally interrupt PLAN80 instead of waiting for the end of output or calculations. For instance, you might start a cash flow application, but remember that you forgot to change a sales assumption. Or half-way though printing a page you might wish you had begun with a different column or row.

From the display options mode or during statement evaluation you may key ! to quit the current file, and # to leave PLAN80. (These keys are used because you must simultaneously press two keys, making it less likely that you will accidentally exit PLAN80.)

PRINTED OUTPUT

Press ESCape to enter the display options mode and key 7 and enter the maximum width of a print line.

Key 8 and enter the maximum number of lines per page.

Turn your printer on. Key P and you will be asked to

```
  Input Destination -->
```

Enter PRINTER or LST and the Example 2 report (including the data changes you made) should appear on your printer. (If you enter any other name your report will be placed on disk as a text file which may be edited and printed.)

Key P again, but this time just press RETURN to print the output on your screen. Note that you have to press the spacebar after each page is printed on your screen.

Also note that printed output includes titles, underlining and blank lines which are excluded when results are displayed.

Key X to leave the display mode. Since no statements follow the DISPLAY statement, you will return to the PLAN80 prompt line.

EXECUTING EXAMPLE 1A

From the PLAN80 prompt line

```
  PLAN80  VERSION 2.3   COPYRIGHT (C) 1982
      BUSINESS PLANNING SYSTEMS, INC.

  INPUT NAME OF PROCEDURE FILE:
    --->
```

enter EX01. Example 1 lets you change any value and recalculate and redisplay results any number of times.

Once in the display mode you should note that 1981 sales and net income are 100 and 18 respectively. Change the 100 to 120 and then press ESCape and +. Model results will be recalculated and net income should have gone to 23. You may want to make other changes and recalculate the results again.

Before leaving the example you may print your results.

If you key ESCape ! you will return to the PLAN80 prompt line, or if you key ESCape # you will exit PLAN80 completely.

RUNNING OTHER EXAMPLES

You should run other PLAN80 examples at this time. Appendix A discusses the output and PLAN80 statements of each example.

LISTING PLAN80 STATEMENTS

PLAN80 will automatically list each statement on your screen as it is processed. To list PLAN80 statements on a printer you should use your editor or CP/M's PIP command.

CREATING NEW APPLICATIONS

To create a new PLAN80 application you use a program, called an editor, in much the same way that you would create a letter or report. Editors come in many varieties and most any one may be used to create a file of PLAN80 statements.

If you have a favorite program for composing letters and reports, then use it to create text files containing PLAN80 statements. Generally, you will find that the ability to move blocks of text, to find and/or change all occurances of one name to another, etc. -- features you appreciate when creating letters and reports -- will be equally appreciated when creating PLAN80 statement files.

It is frequently convenient to use a copy of an old application as the basis for creating a new one.

When developing a new application you may want to name your file TEMP.TXT. Then when PLAN80 asks you for the name of your procedure file you need only press the RETURN key. Don't forget, however, to give your new application a permanent name when it is complete. You can use the CP/M command

```
  A>REN xxxxxx.TXT=TEMP.TXT
```

where xxxxxx is an appropriate designation of your application.

APPENDIX D

PLAN80 ERRORS

WHAT YOU SHOULD DO

If a PLAN80 statement contains errors, such as the name ROW2 when you mean ROW3, there is little PLAN80 can do to help you. You must check that your results are reasonable. It's a good idea to double check the computer's results for at least one column when you first create an application.

Sometimes PLAN80 won't tell you about very obvious errors. For instance, if you omit quote marks from the second line of a row description PLAN80 will assume that you are defining another row.

WHAT PLAN80 WILL DO

When PLAN80 encounters a statement whose syntax is incorrect for any reason then you will get a message. Most messages indicate that PLAN80 was expecting an equal sign, a valid name, etc., but found something else.

Fortunately, PLAN80 will also tell you just where the "something else" was found. You will see a "^" on the line beneath the line containing the error. The error is usually immediately above the "^" or is just to the left. However, you should always look backwards a line or two, because you may have made some other error which was syntacticaly correct, but not what you intended, and which resulted in the error noted. Most of the time you will recognize a misspelled name or keystroke error, or realize that you confused one statement with another.

When PLAN80 does indicate an error you will usually be given the choice of continuing or of leaving PLAN80 so you can use the editor to fix your incorrect statement. If you continue then you may discover other errors, but don't try to salvage results. Correct your input statements and begin again when you have found all your syntax problems.

PLAN80 ERROR CODES

CODE	EXPLANATION
0	A control statement, beginning with ":", expected. Current line ignored.
1	*Invalid control statement. Check spelling.*
2	":COLUMNS" expected. At least one column must be defined prior to statement shown, which will be ignored.
3	":ROWS" expected. At least one row must be defined prior to statement shown, which will be ignored.
4	Too many rows or columns. See Appendix E for limits.
5	Opening quote mark expected.
6	Closing quote mark expected.
7	Opening parenthesis expected.
8	Closing parenthesis expected.
9	Equals sign expected.
10	Operator (+,-,* or /) expected.
11	Invalid option name.
12	"COLUMNS" or "ROWS" expected. You may be missing a control statement such as :RULES or :DATA.
13	Invalid character. Numeric values must begin with a digit, a minus sign or a decimal point.
14	Invalid column or row name.
15	Duplicate column or row name. Each column or row must have a unique name.
16	Diskette not on-line or no space available.
17	Mathematical operator (+,-,*,/ or ^) expected.
18	Continuation of a rule expected.
19	Word "ALL" or opening parenthesis expected.
20	Invalid function name.

CODE EXPLANATION

21 Sum of undiscounted values must be positive.

22 First undiscounted value must be negative.

24 Wrong type. Column name found where row name required or row name found were column name required.

25 Comma expected.

26 ".." expected.

27 Result was defined as a single row/column location. Hence, the expression to the right of the equals sign must represent a single value (not a row or column or expression involving rows and columns.)

28 Limitation on total length of row and column descriptions exceeded. Shorten descriptions.

29 File not found. Check spelling. Also check diskette name and prefix value.

30 Invalid character found where column or row expected.

31 Row name expected.

32 Column name expected.

33 Row or column name expected.

34 Relational operator (=,<>,<,>,<= or >=) expected.

35 Row or column name or single value expression of the form (row name,column name) expected.

36 Extra non-blank characters found at end of line.

APPENDIX E

SIZE CONSIDERATIONS

MAXIMUM NUMER OF COLUMNS AND ROWS

The following table indicates the approximate number of columns and rows which may be defined in a PLAN80 application for a 64k computer. Column A indicates possible numbers for either the column or row dimension, and Column B shows the number to which you are limited in the other dimension. In other words,

* if you have 13 columns then you can have up to 235 rows, or
* if you have 100 rows you can have up to 40 columns, etc.

Column A	Column B
1	511
2	510
3	509
5	452
6	412
10	304
12	269
13	254
15	229
20	183
30	130
40	100
50	81
60	67
70	58
80	50
90	44
100	40
100	40
110	36
120	32
150	25
200	17
250	13
300	10
350	8
400	6
450	5
500	4

The maximum number of rows and columns will be reduced if you use INCLUDE or REPEAT statements.

MODEL SIZE

You can prepare a similar table on your screen indicating the exact limits for your computer by inserting the statement

```
:MODELSIZE
```

as the first or only statement of PLAN80 model. You might want to pencil in a revised Column B above.

TEXT FILE SPACE REQUIREMENTS

Floppy diskette capacities are measured in terms of bytes, the equivalent of a character of test, and range from about 100,000 to 300,000 bytes for 5 1/4" diskettes and 240,000 to 600,000 bytes for 8" diskettes. (For diskettes that use both sides these numbers are doubled.)

Each character of a text file requires one byte. However, a text file does not include all blanks shown on your screen. Included are those upon which the cursor will rest when you space through the file one character at a time. Usually two characters are stored for a RETURN at the end of a line.

Most editors can provide you a measure of a file's size, usually in terms of the total number of characters in the file.

You can also use the CP/M STAT command to determine actual space usage of a file and the total amount of space available on a disk.

IF DISK SPACE IS A PROBLEM

If your computer contains two or more disk drives then the second drive should give you ample flexibility for applications of any size or complexity.

If have only one drive available the following guidelines should help you run most any application.

* Do all editing on a separate disk which contains your editor and then transfer only the files actually needed for your applicaton to a disk which contains only the CP/M system, the PLAN80 system and CONTROL.TRM.

* Send printed output directly to a printer, not to disk. A printed report may be many times larger than the file of PLAN80 statements used to create the report.

* Use the FOR command to limit as much as possible the amount of data sent from one application to another.

* Use the INCLUDE feature to include common data, calculations, rows and/or columns, etc., from a single file.

INDEX