BP.BAT is the main batch file to execute the BEST-PLAN Planning SYSTEM.

TEST.BAT is the batch file to execute the UTILITY program by itself. This program may be used to change disk drive configuration for the SYSTEM. ,e.g. the entire DATA disk contents may be transferred to the hard disk DRIVE C. Presently, the SYSTEM will look for DATA in DRIVE B.

MODIFY0.BAT The user may create MODIFY.BAT, the batch file to contain data editing instructions. This allows the user to use his favorite screen editor/word processor. MODIFY0.BAT shows an example using WORDSTAR. The file is self-explanatory.

BEST-PLAN will check if MODIFY.BAT exists. If it does, then it executes the batch file every time an edit request is made. MODIFY0.BAT should be renamed to MODIFY.BAT if this capability is desired. If MODIFY.BAT does not exist, the supplied online Screen EDITOR is executed. There is an advantage of using the supplied EDITOR, since total system integration is achieved.

RosettaSYSTEMs BEST-PLAN SYSTEM diskette A

```
BP      .BAT      BEST-PLAN main MENU command file
BP-EDT  .EXE      BEST-PLAN Full Screen EDITOR
BP-ORG  .EXE      BEST-PLAN Database Re-organization
BP-PS1  .EXE      BEST-PLAN Optimization Pass 1 datainput
BP-PS2  .EXE      BEST-PLAN Optimization Pass 2 matrixGEN
BP-PS3  .EXE      BEST-PLAN Optimization Pass 3 LP
BP-PS4  .EXE      BEST-PLAN Optimization Pass 4 solution
BP-PSX  .EXE      BEST-PLAN Optimization DIFile input
BP-RPT  .EXE      BEST-PLAN Report/Display Writer
BP-SSV  .EXE      BEST-PLAN Save Solution in database
BP-STR  .EXE      BEST-PLAN Main MENU dispatcher
BP-UTL  .EXE      BEST-PLAN Utility MENU dispatcher
CONTINUE.BAT      ....command file to continue
MODIFY0 .BAT      Command file for USER Editor(WordSTAR)
TEST    .BAT      BEST-PLAN Utility MENU command file
```

RosettaSYSTEMs BEST-PLAN SYSTEM diskette B

BP-COM	.EXE	BEST-PLAN Common executable
BP-ENV	.COM	BEST-PLAN Environment Configuration
BP-ENV	.CON	BEST-PLAN Environment Configuration
BP-PD2	.EXE	BEST-PLAN Optimization Pass 2 d.p.
BP-PD3	.EXE	BEST-PLAN Optimization Pass 3 d.p.
BP-SYS	.EXE	BEST-PLAN Application dispatcher
CONFIG	.SYS	BEST-PLAN config.sys parameter file
LISTME		BEST-PLAN Bill of Material List
README		Readme FILE for instructions

DATA (DEMO) DISKette, 5-1/4 Double density
Double sided

o The DATA (DEMO) DISKette contains all TUTORIAL messages and DATA for six(6) sample problems as discussed in the SAMPLES MANUAL/USERS GUIDE. see manual. The diskette contains two subdirectories, i.e. \DATA and \HELP.

The supplied DATA disk is FULL. The DEMONSTRATION SYSTEM may limit your capability to add scenarios for your "WHAT-IF" analysis. The on-line TUTORIAL messages may be transferred to another drive/directory to facilitate this option.

DEMO Installation: CREATE a sub-directory to contain BEST-PLAN, i.e. BPLAN. Create two(2) sub-directories within BPLAN, i.e. DATA and HELP. Copy contents of the demonstration data disk, contents of A:DATA to C:DATA and A:HELP to C:HELP, respectively. Copy the contents of the two(2) SYSTEM diskettes into the main directory BPLAN. Then use BP.bat to run BEST-PLAN.

```
  RosettaSYSTEMs    BEST-PLAN  DATA   diskette

DATA        <DIR> BEST-PLAN distribution Demo DATA
HELP        <DIR> BEST-PLAN distribution TUTORIAL files

  RosettaSYSTEMs    BEST-PLAN  DEMO DATA files

BP-DEF  .F00      BEST-PLAN default template.f00
BP-TBL            BEST-PLAN Application dispatcher (DEMO)
P1901   .DBS      DEMO1 -Shipping/distribution DATABASE
P1901   .F00      ....DEMO1.  template.00 report
P1901-01.IN       ....DEMO1.  inputDATA  master PLAN
P1901-02.IN       ....DEMO1.  inputDATA  scenario 2
P1901-03.IN       ....DEMO1.  inputDATA  scenario 3
P1902   .DBS      DEMO2 -Diet/Blend  sample    DATABASE
P1902-01.IN       ....DEMO2.  inputDATA  master PLAN
P1902-02.IN       ....DEMO2.  inputDATA  scenario 2
P1903   .DBS      DEMO3 -Multi-period schedule DATABASE
P1903   .F00      ....DEMO3.  template.00 report
P1903-01.IN       ....DEMO3.  inputDATA  master PLAN
P1903-02.IN       ....DEMO3.  inputDATA  scenario 2
P1982   .DBS      DEMO4 -Material Allocation   DATABASE
P1982   .F00      ....DEMO4.  template.00 report
P1982-01.IN       ....DEMO4.  inputDATA  master PLAN
P1982-02.IN       ....DEMO4.  inputDATA  scenario 2
P1983   .DBS      DEMO5 -Energy Planning SystemDATABASE
P1983   .F00      ....DEMO5.  template.00 report
P1983-01.IN       ....DEMO5.  inputDATA  master PLAN
P1983-02.IN       ....DEMO5.  inputDATA  scenario 2
P1984   .DBS      DEMO6 -Production Planning   DATABASE
P1984   .F00      ....DEMO6.  template.00 report
P1984   .F01      ....DEMO6.  template.01 report
P1984   .F02      ....DEMO6.  template.02 report
P1984-01.IN       ....DEMO6.  inputDATA  master PLAN
P1984-02.IN       ....DEMO6.  inputDATA  scenario 2
P1984-03.IN       ....DEMO6.  inputDATA  scenario 3
P1984-04.IN       ....DEMO6.  inputDATA  scenario 4

  RosettaSYSTEMs    BEST-PLAN  TUTORIAL  files

#KEYBD            Tutorial BEST-PLAN Keyboard utilization
#BLANK            Tutorial   default screen blank message
#TUNE             BEST-PLAN music (WilliamTell Overtune)
#MENU             Tutorial LIST for Main MENU dispatcher
#UTLT             Tutorial LIST for Utility MENU
#EDIT             Tutorial LIST for Full Screen EDITOR
#SYSM             Tutorial LIST for Application dispatcher
```

```
  RosettaSYSTEMs    BEST-PLAN  TUTORIALfiles(MENU)

#MENU0  .1       ....Main MENU dispatcher Summary
#MENU0  .2       ....Main MENU dispatcher description
#MENU0  .4       BEST-PLAN system FLOW Diagram
#MENU1  .1       Main MENU select1. report/display
#MENU1  .2       Report/Display template filenames
#MENU1  .3       ....sample Report/Display template
#MENU2  .1       Main MENU select2. List/EDIT input DATA
#MENU2  .2       ....sample inputDATA for a master PLAN
#MENU2  .3       ....sample inputDATA for a scenario
#MENU3  .1       Main MENU select3. Optimization
#MENU3  .2       ....select3. sample inputDATA Error LIST
#MENU3  .3       ....select3. sample RAW Solution LIST
#MENU3  .4       ....select3. sample MATRIX equation LIST
#MENU4  .1       Main MENU select4. Save SOLUTION
#MENU4  .3       ....sample DIFile inputDATA LAYOUT
#MENU5  .1       Main MENU select5. Change Scenario
#MENU5  .2       ....select5. sample scenario report
#MENU6  .1       Main MENU select6. Make scenario as PLAN
#MENU7  .1       Main MENU select7. List/EDIT report tmplt
#MENU7  .2       ReportWriter SUMMARY Description
#MENU7  .3       ReportWriter RULES for Creating Template
#MENU7  .4       ReportWriter special editing/computation
#MENU8  .1       Main MENU select8. Exit to DOS
#MENU9  .1       Main MENU select9.  External Linkages
#MENU9  .2       ....sample DIFile outputDATA LAYOUT
#MENU9  .3       DIFile InputDATA SUMMARY description
#MENU9  .4       DIFile OutputDATA SUMMARY description


  RosettaSYSTEMs    BEST-PLAN  TUTORIALfiles(UTILITY)

#UTLT0  .1       BEST-PLAN Utility MENU LAYOUT
#UTLT0  .2       BEST-PLAN TEST.bat description
#UTLT1  .1       Utility select1. Color settings
#UTLT2  .1       Utility select2. Disk PATHnames
#UTLT3  .1       Utility select3. Optimization constants
#UTLT3  .2       ...LAYOUT of Optimization ReportControl
#UTLT3  .3       ...LAYOUT of Optimization TraceControl
#UTLT3  .4       DIFile IN/OUT parameter description
#UTLT4  .1       DATAEntry SYNTAX Definition SUMMARY
#UTLT5  .1       DATAEntry ERROR Messages SUMMARY
#UTLT8  .1       BEST-PLAN SYSTEM, Other Considerations
#UTLT9  .1       BEST-PLAN SINGLE-STEP optimization
#UTLT9  .2       BEST-PLAN DATABASE Maintenance
```

RosettaSYSTEMs BEST-PLAN TUTORIALfiles(EDITOR)

RosettaSYSTEMs BEST-PLAN TUTORIALfiles(APPLICATIONS)

BEST-PLAN's INSTALLATION and EXECUTION ENVIRONMENT

The SYSTEM (B) DISKette supplied was configured for BOOTSTRAP. BEST-PLAN requires creation or modification of CONFIG.SYS. BEST-PLAN creates a RAMdisk, and maintains memory-resident routines to support the screen displays and the sparse matrix arithmetic. This BOOTSTRAP disk must be created with your copy of the IBM-PC DOS Operating System. It must contain the following files:

- o DOS COMMAND.COM from DOS system diskette.
- o BEST-PLAN's SYSTEM (B) DISKette files; CONFIG.SYS and BP-ENV.COM.

BEST-PLAN also uses INTERRUPT 44(hex) to maintain memory-resident routines. Possible installation ERRORs are:

- o ..BEST-PLANs CONFIG.SYS ..not ACTIVATED!!
- o ..BEST-PLANs INITIALIZATION error, bad INSTALL!!
- o ..BEST-PLANs BP-ENV.CON ..not FOUND!!

INSTALLATION FOR IBM-PC/XT/AT with HARD-DISK BOOTSTRAP:
BOOTSTRAPPING the BEST-PLAN Environment will install the BEST-PLAN device driver for a VDISK type RAMdisk. BEST-PLAN emulates a VDISK(v2.0) driver for compatibility. The supplied CONFIG.SYS contains the line:

```
DEVICE=A:\BP-ENV.COM /S=160 (or /X=160) /B=2
```

where: BP-ENV.COM is the BEST-PLAN RAMdisk device driver. (no spaces around the "=")
/S=mmm, where mmm=the SIZE of RAMdisk (in 1K bytes). i.e. VDISK.
/B=nn, where nn=the number of BUFFERS(4 kbytes) to be allocated for the SPARSE matrix arithmetic.

NOTE: Change /S to /X for extended memory:
/X=mmm, where mmm=the SIZE of EXTENDED MEMORY RAMdisk (in 1K bytes), above 1 megabyte.

This driver must be installed LAST if your SYSTEM is already setup to support OTHER device drivers. ADD THIS LINE TO YOUR CONFIG.SYS, and it should be the LAST "DEVICE=" line. Also set FILES=20, and BUFFERS= will be your choice.

INSTALLATION PROCEDURE:

The BEST-PLAN PACKAGE may be INSTALLED with the following procedure:

IBM-PC without HARD DISK

The BEST-PLAN System program files CANNOT fit a single DISKette at 360Kbytes capacity. Some options must be eliminated to do so, i.e. double precision arithmetic options (BP-PD2.exe and BP-PD3.exe). Care must be exercised in doing this. Copy all files named BP-???.exe into one(1) DISKette and also the batch files (*.bat) and BP-ENV.con.

- o Create the BOOTSTRAP DISKette using IBM-DOS utility FORMAT.COM and SYS.COM as mentioned in the previous page, or create the necessary CONFIG.SYS, adding the line, DEVICE=BP-ENV.COM /S=mmm /B=nn.

- o BOOTSTRAP the SYSTEM (alt-ctl-del) with the newly created BOOTSTRAP DISKette, and then replace it with the BEST-PLAN SYSTEM disk in DRIVE A. The DEMO DATA DISKette must be in DRIVE B to start the DEMO package. Run TEST.bat and change DISK PATHnames for data, Selection 2. to refer to DRIVE B. (see UTILITY section)

- o For users who want to start up a NEW application other than the ones supplied, DUPlicate (DISKCOPY) the supplied Demonstration DISKette unto another diskette, and delete the contents of the \DATA subdirectory. This will allow the creation of a new application plus the availability of the HELP facility. This diskette with an EMPTY \DATA subdirectory can be DUPlicated any number of times as needed per application.

The file "BP-TBL" is a parameter control table discussed in the UTILITY section of the documentation. It allows the user to create an application dispatcher. Up to ten (10) applications can be activated from the dispatcher using the built-in MENU (see. BP-TBL) The number of applications is physically limited to the size of problems since they all must reside in the same DATA DISKette. Files with extensions ".SSS" are session files for BEST-PLAN to remember the last analysis session. This will be activated on entry to that application.

One or more of the supplied SAMPLE applications may apply to your specific requirements. All DATA files for a specific application start with the same PREFIX, i.e. for application PLAN=1984, COPY P1984*.* will transfer all data files for the production scheduling problem unto a new DATA DISKette.

You may have to CHANGE the CRT (screen) attribute assignments using the TEST.BAT procedure. (see UTILITY section of the SAMPLES MANUAL/USERS GUIDE.

IBM-PC/XT/AT Configurations
IBM-PC with HARD DISK

To INSTALL the BEST-PLAN system unto a IBM-PC/XT or an IBM-PC with a HARD DISK, follow the following procedure:

- Transfer contents of the SYSTEM (A&B) DISKettes to the HARD DISK. A subdirectory may be created to incorporate all program and system files of BEST-PLAN. i.e. \BPLAN may be created to contain contents of the supplied SYSTEM (A&B) DISKettes. Subsequent execution of BEST-PLAN will now be initiated in this directory.

- Create the BOOTSTRAP DISKette using IBM-DOS utility FORMAT.COM and SYS.COM as mentioned in the previous pages, or create the necessary CONFIG.SYS, adding the line, DEVICE=BP-ENV.COM /S=mmm /B=nn.

BEST-PLAN's BP-ENV.COM automatically creates the working environment for BEST-PLAN including the creation of an electronic (RAM) disk, by emulating VDISK v2.0. The electronic (RAM) disk will be given the next available drive letter.,e.g. for a two floppy system it will be DRIVE C, and DRIVE D for the mentioned PC/XT.

Now copy the contents of the DEMO DATA DISKette into the HARD DISK by creating two(2) subdirectories within BPLAN called DATA and HELP respectively. The DATA DISKette is supplied with two (2) subdirectories, i.e. \DATA and \HELP. Copy the contents of the DATA DISKette into the HARD DISK in their corresponding subdirectories. If you like to modify the subdirectory names, BEST-PLAN should be properly notified of this change. EXECUTE "TEST.BAT" and select 2 (Disk PATHnames for DATA/SYSTEM files) from the UTILITY MENU. Then indicate where the DATA and HELP files are transferred.

- o For users who want to start up a NEW application other than the ones supplied, specify which subdirectory will contain the DATA files by the execution of TEST.BAT.

One or more of the supplied SAMPLE applications may apply to your specific requirements. All DATA files for a specific application start with the same PREFIX, i.e. for application PLAN=1984, COPY P1984*.* will transfer all data files for the production scheduling problem into a new DATA subdirectory.

The file "BP-TBL" is a parameter control table discussed in the UTILITY section of the documentation. It allows the user to create an application dispatcher. Up to ten (10) applications can be activated from the dispatcher using the built-in MENU (see. BP-TBL) The number of applications is physically limited to the size of problems since they all must reside in the same storage device. Files with extensions ".SSS" are session files for BEST-PLAN to remember the last analysis session. This will be activated on entry to that application.

You may have to CHANGE the CRT (screen) attribute assignments using the TEST.BAT procedure. (see UTILITY section of the SAMPLES MANUAL/USERS GUIDE.

DEMONSTRATION SYSTEM Applications MENU

```
╔══════════════════════════════════════════════════════════════╗
║          BEST-PLAN  Sample Demonstration System              ║
╠══════════════════════════════════════════════════════════════╣
║                                                              ║
║  ..Mathematical Formulation Examples                         ║
║   1.       PLAN: 1901   small Distribution Model             ║
║   2.       PLAN: 1902   Animal Feed (Diet) Model             ║
║   3.       PLAN: 1903   Multi-period scheduling              ║
║  ..STEP-by-STEP Flow Representation Examples                 ║
║   4.    PLAN: 1982   Raw Material Allocation Model           ║
║   5.    PLAN: 1983   Energy Planning System                  ║
║   6.    PLAN: 1984   Production Planning Model               ║
║               ................................               ║
║       Up to nine(9) PLANs may be supported by this           ║
║       application DISPATCHER  capability. (BP-TBL)           ║
║               ................................               ║
║   0.      .......EXIT back to  P C  D O S.......             ║
║                                                              ║
╚════════════════ PRESS (alt) and the Number ══════════════════╝
```

Use the ALT-key and the number to select a demonstration application. Or simply select a PLAN by typing the four(4) digit plan number, as a response to the PROMPT ..PLANselect:.

The DEMONSTRATION SYSTEM Application MENU can be by-passed by simply starting BEST-PLAN as follows:

```
╔═══════════════════════════════════════════════════╗
║                                                   ║
║      BP nnnn      where nnnn= the PLAN number     ║
║                                                   ║
╚═══════════════════════════════════════════════════╝
```

BEST-PLAN OPTIMAL PLANNING SYSTEM. HOW TO GET STARTED

══════ A WALK THROUGH ══════

a. Please take the time to read the USERS GUIDE MANUAL.

MENU	section discusses the disciplines of in running the SYSTEM.
UTILITY	section discusses how to SETUP your own CUSTOM tailored working ENVIRONMENT.
REPORT	section discusses how to create your own report or change what is already supplied.
EDITOR	section discusses how to use the online Screen EDITOR.

b. Please read the INSTALLATION instructions as discussed in this section. Three(3) batch FILES may be modified to suit your own directory structure, namely; BP.BAT, TEST.BAT and MODIFY.BAT.

c. Execute TEST.BAT to setup PATHname parameters for your DATA files, as well as possible CRT (screen) attributes for Color/Contrast.

d. Execute BP.BAT to startup the BEST-PLAN SYSTEM. The Application DISPATCHER MENU will show the six(6) applications included in the DATA DISKette. SELECT any one of these applications using the (Alt)-n, where n=the MENU number. The system will initialize at the last known session for that PLAN, and present the startup report/display (TEMPLATE=0). The ON/OFF switch for the ONLINE TUTORIAL is (Alt)-H. At this point the ONLINE TUTORIAL will give you a brief description of the six(6) sample problems supplied with the DEMO DISKette.

Sample SESSION: While in the Application DISPATCHER MENU

1. Depress (Alt)-1 .selection 1.

PLAN:1901 Shipping Distribution Problem.

The SYSTEM will display the last known session on this PLAN. You will be looking at a typical report/display (TEMPLATE=0) presented with the solution values for the scenario. (SCENARIO=1)

2. Depress (Alt)-2 .selection 2.

Request for a LISTING of the input data. Put this display into screen 4. The SYSTEM will display the input data for this MODEL (SCENARIO=1). This was created using the supplied Screen EDITOR. Depress function-key 1 (F1), the SYSTEM displays our first report. You can flip between displays by depressing the necessary function key.(i.e. F4 is screen 4.) Do this same operation again, but request the LISTING of input data for scenarios 2 and 3. Display them using screens 5 and 6, respectively.

3. Depress (Alt)-1 .selection 1.

Request for report/display using TEMPLATE 0, and display the report on screen 2 for scenario 2. The SYSTEM will display the solution values of SCENARIO=2 (WHAT-IF) using the same template as our initial report. Again you can flip between screen 1 and screen 2 and notice the change in the BEST shipping plan for minimized COST. DO the same thing for SCENARIO=3, and display the report on screen 3.

4. Depress (Alt)-7 .selection 7.

Request for a display ONLY of TEMPLATE 0, and put the display on screen 8. The SYSTEM will show the report/display template created to produce the display. Again, this was created using the supplied Screen EDITOR or any standard screen editor (non - DOCUMENT ,ASCII).

5. At this point, Depress (F9) .function-key 9.

The Table of Contents for all eight(8) screen will be displayed at the upper left corner of screen 1. This function-key (F9) acts as a switch to turn the display ON or OFF. It will be noted that all eight(8) screens are loaded with their corresponding displays. The function-keys (F1)-(F8) may be used to flip from one display to another. Screen 1, will carry all the pop-up MENUs.

6. Depress (F10) .function-key 10.

The main MENU will pop-up at the upper right corner of screen 1. This function-key (F10) acts as a switch to turn the main MENU display ON or OFF. Function-key (F10) is also used to ABORT! execution of any of the sub-programs and re-initializes the current MENU.

7. Depress (Alt)-H .HELP request.

The tutorial will be presented using screens 1, 2, 3 and 4. This key (Alt)-H acts as a switch to turn the HELP facility ON or OFF. The previous displays on screen 1, 2, 3 and 4 will be re-displayed.

8. Depress (Alt)-T .TIME-of-Day request.

The Time-of-day can be displayed at the upper right corner of the Screen 1. using (Alt)-T. This will change to seconds to complete optimization, during the optimization process.

9. The screen controls are operational while viewing a particular screen. PAGE-UP and PAGE-DOWN control the screens 22 lines in both directions. CURSOR-UP and CURSOR-DOWN control the screens 6 lines in both directions. HOME restores the beginning of the display. (Alt)-P will print the currently displayed screen. It you have a print spooling program, install it to improve system performance.

BEST-PLAN is completely menu driven, and guides the planner through the course of the interactive session using STEP-BY-STEP informational messages. Just like any other planning system, BEST-PLAN assumes that the problem is well defined and data properly collected and evaluated for accuracy.

ANALYSIS TERMINOLOGY

- MASTER PLAN or problem model will be assigned a number 1-9999. The four digit capability is to allow the planner to incorporate a possible month-year combination to identify a PLAN. e.g. PLAN 1082 may mean a production plan for October, 1982.

The PLAN may be created using any available screen editor or word processor. The INPUT format of the PLAN data is totally FREE-FORM thus giving the planner flexibility and freedom in constructing the model. Comment lines must be generously included in the Plan for self-documentation. The planner must be able to relate to the PLAN for future reference.

The MASTER PLAN is given the

```
FILENAME: Pnnnn-01.IN
           nnnn = PLAN NO.(1-9999)
```

- SCENARIO NO. or (what-ifs) analysis will also be assigned a number 1-25. Scenario 01 is assigned to the master plan, scenario 02-25 are assigned to any corresponding analysis to be made relative to this plan.

SCENARIO Update may be created using any screen editor or word processor available. It follows the same INPUT format as the PLAN, except that only the changes to the PLAN are incorporated in the file.

The SCENARIO updates are given the

```
FILENAME: Pnnnn-ss.IN
           nnnn = PLAN NO.(1-9999)
             ss = SCENARIO NO.(2-25)
```

- REPORT Templates will be created for this PLAN. They will be assigned a number 0-99. Report Template.00 will be the main menu template, which is automatically displayed on start-up. A default MENU (BP-DEF.F00) is used when the Report Template.00 is not available. When Template.00 does not exist, the system may use the INPUT data Pnnnn-01.IN as Template.00 replacement, IF and ONLY IF the first line contains the .DF (default field definition) directive. This allows the user to create an INPUT file which also works as the REPORT file to superimpose input and solution. The user must be responsible for managing such combination using input comment lines. These report templates are also created using any screen editor or word processor available. They all follow the general rules concerning report generation. (see. Report)

The REPORT templates are given the

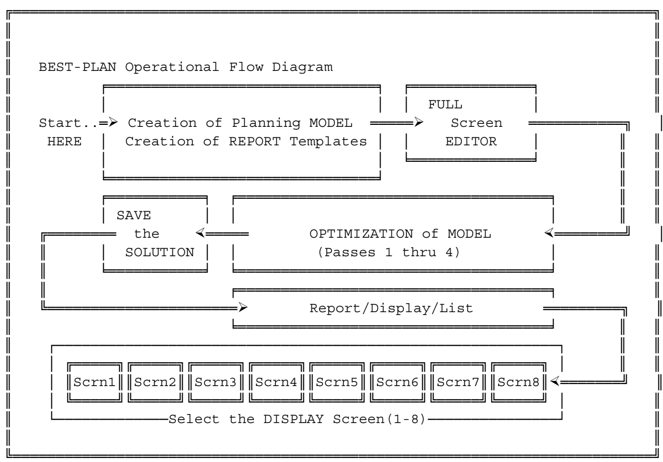
```
FILENAME: Pnnnn.Frr
          nnnn = PLAN no.
            rr = report template no.
```

REPORT templates must be located in the same disk/directory as the master PLAN.

SAMPLE REPORT Template.00 (MAIN MENU) Pnnnn.F00

```
.df=vl.e34
 ╔════════════════════════════════════════════════════════════════════╗
 ║               A.B.C. OIL CO.  PLANNING SYSTEM                      ║
 ║ |title                                                         |   ║
 ╠═══════════════════════════╦════════════════════════════════════════╣
 ║ plan:|plan| scenario:|sn| ║  last analysis:  |sndate| |sntime|     ║
 ╠═══════════════════════════╣  cost(minimized) |optimal=t$.e42 |     ║
 ║ selections:               ║ infeasibilities  |inf=vl|              ║
 ║ 1. report writing         ╠════════════════════════════════════════╣
 ║ 2. update/data entry      ║        master plan   this scenario     ║
 ║ 3. ..OPTIMIZATION...      ║..PLT#1 production:                     ║
 ║ 4. save NEW solution      ║ D10   |#210.s1.t1  |  |#210.t2     |   ║
 ║ 5. change/remove scenario ║ D30   |#230.s1.t1  |  |#230.t2     |   ║
 ║ 6. make scenario->MASTER  ║ D60   |#260.s1.t1  |  |#260.t2     |   ║
 ║ 7. edit template(REPORT)  ║ total=|t1          |  |t2          |   ║
 ║ 8. EXIT to IBM-DOS        ║..PLT#2 production:                     ║
 ║ 9. special INTERFACES     ║ M10   |#510.s1.t3  |  |#510.t4     |   ║
 ╠═══════════════════════════╣ M30   |#530.s1.t3  |  |#530.t4     |   ║
 ║    The A.B.C. Company     ║ M60   |#560.s1.t3  |  |#560.t4     |   ║
 ║    scheduling depart.     ║ total=|t3          |  |t4          |   ║
 ╚═══════════════════════════╩════════════════════════════════════════╝
```

BEST-PLAN OPERATIONAL DIAGRAM



OPERATIONAL PROCEDURE

The problem is transcribed into a model using either one of two methods suggested or a combination of both methods. (see. Modeling Methods). A PLAN NO. is assigned to this problem model.

The model may be entered using:

1. STEP-BY-STEP Material FLOW Representation Method.
2. Mathematical Formulation or Equations

REPORT templates may be created for analysis and display of solutions. Report template.00 represents the MASTER MENU which is automatically displayed on SCREEN 1.

SELECTING AN OPERATION

The MENU selections are accomplished by depressing both the ALT key and the desired number. This is to protect the session from inadvertent key depression. TUTORIAL information for these operations are available by switching ON the TUTORIAL (ALT)-H, then selecting the operation for which the TUTORIAL is desired. TUTORIAL is ONLY applicable when the HELP tutorial is accessible. There are 81 files in the DEMO DATA DISKette to support the online TUTORIAL. These files must be accessible through the proper entry in Selection 2., (Disk Pathnames and Subdirectories) of the UTILITY MENU (see. UTILITY).

- o SCREEN 1 represents the current scenario. It may contain a report using TEMPLATE.00, BP-DEF.F00, or the INPUT DATA list as the case maybe.
- o FUNCTION KEY F9 is used to switch on/off the Table of Contents of SCREENs 1-8. The Table of Contents is presented at the upper left corner of SCREEN 1.
- o FUNCTION KEY F10 is used to switch on/off the display of the main pop-up MENU. The main MENU is displayed at the upper right corner of SCREEN 1. F10 is also used to ABORT current operation and return back to main MENU.

```
╔═════════ MAIN MENU ═════════╗
│ 1. Report using TEMPLATEs   │
│ 2. EDIT/LIST InputDATA      │
│ 3. OPTIMIZATION             │
│ 4. SAVE the Solution        │
│ 5. Change/Delete Scenario   │
│ 6. PROMOTE Current Scenario │
│ 7. EDIT/LIST TEMPLATEs      │
│ 8. Exit back to MS-DOS      │
│ 9. PERFORM a (DOS) Shell    │
│ 0. Utility MENU Operations  │
╚═══════ KEYs(Alt)-No. ═══════╝
```

SCREEN 1: MAIN MENU OPERATION

The main MENU screen is utilized as follows:

> Display-Line.01 is used to confirm the selected menu procedure, plus any additional operational messages.
>
> Display-Line.25 is used for informational messages during the course of the interactive session.
>
> A DATA Entry window is also created at the lower left bottom of SCREEN 1 for the interactive questions.

Both Display-Line.01 and Display-Line.25 are displayed using the banner colors.

```
╔═══════════════════════════════════════════════════════════════════════════╗
║┌─────────────────────────────────────────────────────────╥────────╥──────┐║
║│ DISPLAY Line 1      OTHER MESSAGES ( HEADER )            ║  TIME  ║      │║
║├─────────────────────────────────────────────────────────╨────────╨──────┤║
║│╒══════════════ TABLE of CONTENTS ══════════════╕╒═════════ MAIN MENU ════════╕│║
║││ 1.                                            ││ 1. Report using TEMPLATEs  ││║
║││ 2.                                            ││ 2. EDIT/LIST InputDATA     ││║
║││ 3.            ┌────────────────────┐          ││ 3. OPTIMIZ┌─────────────┐  ││║
║││ 4.            │ for all SCREENs (1-8) │       ││ 4. SAVE th│ MENU WINDOW │  ││║
║││ 5.            └────────────────────┘          ││ 5. Change/└─────────────┘  ││║
║││ 6.                                            ││ 6. PROMOTE Current Scenario││║
║││ 7.                                            ││ 7. EDIT/LIST TEMPLATEs     ││║
║││ 8.                                            ││ 8. Exit back to MS-DOS     ││║
║│╘══════════════ USE KEYs(F1)-(F8) ══════════════╛│ 9. PERFORM a (DOS) Shell   ││║
║│                                                 │ 0. Utility MENU Operations ││║
║│                                                 ╘═════════ KEYs(Alt)-No. ════╛│║
║│  ╔═══════════════════════════════════════════════════╗                        │║
║│  ║                                                   ║                        │║
║│  ║                                                   ║                        │║
║│  ║                                                   ║                        │║
║│  ║                                                   ║                        │║
║│  ║          ┌────────────────────┐                   ║                        │║
║│  ║          │ DATA ENTRY WINDOW  │                   ║                        │║
║│  ║          └────────────────────┘                   ║                        │║
║│  ║                                                   ║                        │║
║│  ║                                                   ║                        │║
║│  ║                                                   ║                        │║
║│  ╚═══════════════════════════════════════════════════╝                        │║
║├──────────────────────────────────────────────────────────────────────────────┤║
║│ DISPLAY Line 25        INFORMATIONAL MESSAGES                                │║
║└──────────────────────────────────────────────────────────────────────────────┘║
╚═══════════════════════════════════════════════════════════════════════════════╝
```

OTHER CONSIDERATIONS

Scenario data files are established as mere updates to the master plan. Each scenario, including the PLAN will carry its own unique work files and matrix (table of equations). A scenario analysis does not actually change the PLAN. Changes are made to a temporary file prior to optimization.

All reports and displays are relative to solutions SAVED into the solution database. Since the solution database carry all solutions for all scenarios, comparing scenarios is as easy as creating a report template. The solution database is a randomly accessible datafile with a HASHED key-index table.

The PLANNING DATABASE is given the

```
FILENAME: Pnnnn.DBS
            nnnn = PLAN no.
```

The solution database is initialized during start-up if the file Pnnnn.DBS does not exist. Databases, as well as database dumpfiles are created in the same disk/directory as the master PLAN files.

SCREEN DISPLAYS for REPORTS

BEST-PLAN allows eight (8) independent display screens. All reports may be displayed in any one of the available screens. Report template.00, the MAIN MENU is always displayed on SCREEN 1.

With a color monitor, all screens will carry their own unique color assignments. These color assignments may be modified using the UTILITY procedure. (see. UTILITY)

SCREEN CONTROLS

The FUNCTION KEYS F1 to F8 is conveniently used to switch from one display screen to another. Once displayed, the report or display can be manipulated using the following:

```
up-arrow          scroll the report down    6 LINES
down-arrow  scroll the report up      6 LINES
pageUp            move display a page up    22 LINES
pageDN            move display a page down  22 LINES
Home              display first page
Alt-P       print displayed report (LPT1:)
Alt-T       Timer(clock) ON/OFF switch
F1-F8       display another screen
```

```
 Scrn1  Scrn2   The Function Keys are used as follows:
                    F1-F8  screen displays. Up to 8 independent
                           screens, flip from one to another.
 Scrn3  Scrn4          F9  Table of Contents for the screens
                      F10  ABORT/MENU ON/OFF switch

 Scrn5  Scrn6                                        home   up.6   pg.up

 Scrn7  Scrn8              The Numeric Keypads are:
                           home   display page 1
                           ↑      up   by 6  lines
                           ↓      down by 6  lines
  TOC   abort              pgup   up   by 22 lines          dn.6   pg.dn
                           pgdn   down by 22 lines
Function Keys                                        Numeric Keypads
  LEFT HAND                                            RIGHT HAND
```

KEYBOARD LAYOUT

The figure show the keyboard layout of the BEST-PLAN System.

(Alt)-H	can be used to activate/deactivate the online TUTORIAL.
(Alt)-P	can be used to print the currently displayed SCREEN.
(Alt)-T	can be used to activate the TIMER/CLOCK display on the right-hand corner of SCREEN 1. This display changes to SECs to complete during an OPTIMIZATION run.

```
[ 1 ] [ 2 ] [ 3 ] [ 4 ] [ 5 ] [ 6 ] [ 7 ] [ 8 ] [ 9 ] [ 0 ]

     [ Alt ]  Selections are made with the (Alt)-"N"  N=number
              If MENU ON, up(↑)-down(↓) to select, ENTER to activate


[Scrn1][Scrn2]  The Function Keys are used as follows:
                   F1-F8  screen displays. Up to 8 independent
                          screens, flip from one to another.
[Scrn3][Scrn4]        F9  Table of Contents for the screens
                     F10  ABORT/MENU ON/OFF switch
                  (Alt)-P  Print screen LPT1:
[Scrn5][Scrn6]    (Alt)-T  ClockTimer ON/OFF switch    [home ][up.6 ][pg.up]
                  (Alt)-H  Tutorial   ON/OFF switch

[Scrn7][Scrn8]            The Numeric Keypads are:     [music][     ][     ]
                          home   display page 1
                           ↑     up   by 6  lines
[ TOC ][abort]             ↓     down by 6  lines      [     ][dn.6 ][pg.dn]
                          pgup   up   by 22 lines
                          pgdn   down by 22 lines
Function Keys              <-     music(if requested)    Numeric Keypads
  LEFT HAND                                               RIGHT HAND
```

SUMMARY OF MENU SELECTIONS

1. Report or Display. A report template is displayed or printed on the line printer. The report template is specified and displayed on the screen (or line printer) where the report is to be presented.

2. Data Entry or Update. BEST-PLAN activates the user's screen editor to allow the planner to either modify or create a scenario analysis datafile.

3. OPTIMIZATION. The optimization is activated, producing an optimized solution in temporary file.

4. SAVE Solution. The previously optimized solution is saved into the solution database for reporting and analysis.

5. CHANGE Scenario. Current analysis is transferred to another scenario or "what-if"analysis.

6. MAKE this Scenario the PLAN. Allows the planner to modify the master PLAN. This operation will install current scenario as the NEW master PLAN, thus allowing for progressive SOPHISTICATION.

7. EDIT a Report Template. BEST-PLAN activates the user's screen editor to allow the planner to modify or create a report template.

8. EXIT to PC-DOS. Formal exit to the operating system. DONE with the ANALYSIS.

9. Perform a (Dos) Shell command. Allows BEST-PLAN to activate another foreign program or process (e.g. VISICALC, LOTUS 1-2-3, etc.) and to return back to BEST-PLAN when done.

10. BEST-PLAN UTILITY. A utility program is supplied with the system to allow the planner to control system operation and working environment.

SELECTION 1. REPORT/DISPLAY using the REPORT program

The report program requires the following additional information:

```
          ..Display DATA Dictionary :  ? Yes/No
                      if ONLY the DATA Dictionary is needed
           ..Reporting for Scenario :  ? NN = 0-25 scenarios
                                         defaults to current
        ..enter REPORT template no. :  ? NN = 0-99 templates
                                     or  N ..use Pnnnn-ss.IN?
        ..option SCREEN=(1-8),LPT=9 :  ? SS
                                         SS = 1-9  option
```

The planner can select any of the eight (8) screens available to display the required report. This operation allows the planner to have up to eight (8) independent reports (outstanding) display able at the same time. The planner may then flip from one report to another using the FUNCTION KEYS F1 to F8.

The cursor control characters (e.g. up-arrow and down-arrow) may be used to scroll the report up or down.

The (Alt)-P can be used to transfer the SCREEN (1-8) currently on display to the line printer (LPT1:)

The DATA Dictionary requested will contain the LISTING of processes' and resources' numbers and their corresponding NAME IDENTIFICATION (30-character description). The format is as follows:

```
         ............master plan:P1901.DBS............


            ......P R O C E S S E S - V E C T O R S......
   @2          =Jamesville
   @1          =Flint
   @3          =St.Louis
   @3 #100     =Product AA(St.Louis)
   @1 #100     =Product AA(Flint)
   #3          =Dayton demand
   #100        =Product AA
```

The main Report Template is Pnnnn.F00, where nnnn= the PLAN number. It was the intention that this template be used as a quicklook facility to query the Scenario's solution during the START of an ANALYSIS. If non-existent, the PLAN's INPUT DATA file can be the report template or the default supplied. (BP-DEF.F00). Using the PLAN's INPUT DATA file allows the planner to combine a REPORT template with INPUT DATA or vice-versa, a rather interesting effect. e.g. P1902-01.IN The single quote (') makes the line transparent to the INPUT SYNTAX processor while allowing it to be processed by the REPORT generator. The first line must be the .DF (default field definition) directive for the system to recognize this intention.

```
'.df=vl.e31 <-----------------MUST BE THE FIRST LINE-------
'.d1=0.2*(#1)+0.45*(#2)+0.1*(#3)+0.6*(#4)+0.5*(#5)+0.3*(#6)
'.d2=0.5*(#1)+0.25*(#2)+0.2*(#3)+0.2*(#4)+0.1*(#5)+0.2*(#6)
'.d3=0.3*(#1)+0.30*(#2)+0.7*(#3)+0.2*(#4)+0.4*(#5)+0.5*(#6)

'     |title                                                   |

COSTS    #1=0.10/barley,              #2=0.18/corn
         #3=0.06/oats,                #4=0.14/rice
         #5=0.22/meat byprods.,       #6=0.18/bones;

        EQUATIONS

   totalw    #1   +   #2   +   #3   +   #4   +   #5   +   #6  <EQ> 100.;
'      (|#1.t1 |)(|#2.t1|)(|#3.t1|)(|#4.t1|)(|#5.t1|)(|#6.t1 |)
'       total COSTS= |optimal.t$.e42|         total weight=|t1      |

       com.b1     0.2*#1 + 0.45*#2 + 0.1*#3 + 0.6*#4 + 0.5*#5 + 0.3*#6
                  <RN> 10.0:30.0; 'Component B1  |fn=d1 |

       com.b2     0.5*#1 + 0.25*#2 + 0.2*#3 + 0.2*#4 + 0.1*#5 + 0.2*#6
                  <RN> 20.0:40.0; 'Component B2  |fn=d2 |

       com.b3     0.3*#1 + 0.3*#2 + 0.7*#3 + 0.2*#4 + 0.4*#5 + 0.5*#6
                  <RN> 20.0:35.0; 'Component B3; |fn=d3 |
```

SELECTION 2. DATA ENTRY using available SCREEN EDITOR

Data entry is accomplished using the supplied screen EDITOR. The Editor has a repertoire of control statements, called directives by means the user (planner) can invoke a wide range of editing functions.

Since everybody have their own favorite screen editor, the system also allows the planner to link his favorite screen editor using the procedure in the installation section. (see. EDITOR section)

SELECTION 3. OPTIMIZATION

Selection 3 processes the input data, master plan or a scenario as the case may be, creates the database (or updates to the database) and creates the mathematical model (sets of equations) for optimization. During the optimization, BEST-PLAN may display the following:

```
      The INPUTdata ErrorLIST             on SCREEN 2.
          includes errors detected
      The "RAW" SOLUTION LIST             on SCREEN 3.
          includes "trace" messages
      The Matrix (EQUATIONS)                    on SCREEN 4.
          entered/or formulated by BEST-PLAN
```

Screens 3 and 4 may be individually enabled/disabled using the UTILITY operation. The optimization will be terminated if errors are detected in the INPUT data.

Informational messages are periodically displayed on Display.LINE.1 & .25 of SCREEN 1. The time-to-finish for optimization will be on the right-hand-corner on LINE.1 if the (Alt)-T is switched ON.

The performance during optimization may be controlled by the planner by setting some constants and conditions using the UTILITY operation. (see. SELECTION 10. UTILITY) The following are possibles:

changing optimization constants
changing the number to estimate time-to-finish
changing the default objective function
enable/disable the above mentioned standard displays
enable/disable cost propagation after optimization
tracing the iterations and report messages

There will be four (4) possible outcome during the optimization:

1. Error detected during the processing of INPUT data. Normally known as SYNTAX errors, the INPUT data LIST displayed on SCREEN 2 will show errors as they are detected. (see. ERROR Messages). Please correct errors and resume analysis.

2. Problem is unbounded. Initial processing of the data shows that a resource or a variable is not properly bounded or limited. This means that the mathematics will produce a solution equal to INFINITY which is indeterminate or not VALID. Check the flow of model, or the equations and properly limit the unbounded variable. BEST-PLAN will attempt to locate the unbounded variable or resource for the planner.

3. Problem is infeasible. Infeasibility occurs when one or more of the constraints (LIMITS) cannot be satisfied. This may be due to the limited amount of materials or resources to satisfy the demand. INFEASIBLE SOLUTION MAY BE AN ACCEPTABLE SOLUTION. An Infeasibility LIST is created for the planner on SCREEN 3. This includes the list of resources or variables affected by the infeasibilities and the corresponding equations not properly satisfied.

4. Problem is OPTIMAL. An optimal solution has been detected with all the constraints (LIMITS) satisfied. A "RAW" Solution LIST may be created for the planner on SCREEN 3.

DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Processes three(3) oil based
products through 2 plants.

Excerpt of the "RAW" SOLUTION LIST is shown:

```
     ..........master plan no:  1984   scenario no:  4 ...........


              *** R A W   S O L U T I O N (LIST) ***


     ......S O L U T I O N ( INFEAS. )......

          max: profit   INFEASIBLE       311.5366E-02
                         ..TOTAL...      *************

           THE FOLLOWING ARE AFFECTED BY THE INFEASIBILITIES

......NAME.........ACTIVITY......LOWER LIMIT....UPPER LIMIT..........

#560               8.855767       0             ..none..
#660               0              0              25
#661               12             12             12
#760               24             24             24

 ROW:   e@25 #660      <EQ>  RHS= -3.144233
    + 1*(#560) - 1*(#660) + 1*(#661) - 1*(#760)

.......................................................................

     ......S O L U T I O N ( INFEAS. )......


          max: profit   FUNCTIONAL      -200.8759E+05
                        OBJECTIVE       *************

          ITERATIONS:   36              OPTIMIZATION DATE: 05-01-1988
     INFEASIBILITIES:    1                           TIME: 12:00:00
```

```
V A R I A B L E S (COLUMNS)

 ......................................................................
     ACTIVITY :  VL= volume   R$= reduced cost   I$= internal cost
                 P$= price    LL= lower limit    UL= upper limit
 ......NAME....AT................ACTIVITY.............................

 #10         BS VL= 29.99997  UL= 30
 #100        UL VL= 900  I$=-25000  P$=-25000  FX= 900
 #11         BS VL= 1800000  I$=-1  P$=-1
 #20         UL VL= 30  R$= 2.028242  UL= 30
 #21         BS VL= 105903.8  I$=-1  P$=-1
 #210        BS VL= 90  I$= 270000
 #230        BS VL= 37.29291  I$= 270000
 #330        BS VL= 84.44444  I$=-270000
 #52         BS VL= 90  I$= 270000  P$= 22000
 #53         BS VL= 86.6506  I$= 552431.6  P$= 22000
 #530        BS VL= 38  I$= 602666.6
 #560        ** VL= 8.855767  I$= 695384.8
 #610        UL VL= 25  R$= .2058906  I$=-438387.1  UL= 25
 #611        UL VL= 15  R$= .5452108  FX= 15
 #630        BS VL= 20  I$=-603999.9  UL= 25
 #710        LL VL= 33  R$= .5452108  I$=-438387.1  FX= 33
 #730        LL VL= 33  R$= .7511014  I$=-603999.9  FX= 33
 #760        ** VL= 24  R$= 1  I$=-696923.3  FX= 24
 @10 #100    BS VL= 372.9291  I$= 27000  UL= 900
 @11 #100    BS VL= 527.0709  I$= 27000  UL= 900
 @20 #310    BS VL= 69.35484  I$= 270600
 ......................................................................
```

- process No. and resource No. (@ppp #nnnn)
  ppp=process no.
  nnnn=resource no.

- resource information

price	P$=value
lower limit	LL=value
upper limit	UL=value
fixed limit	FX=value

- solution data as computed by BEST-PLAN

volume	VL=value
Internal Cost	I$=value
Reduced Cost	R$=value

	ERROR MESSAGES	SUGGESTED REMEDY
1.	Processes are assigned nos. 1-999 ONLY	change the No.
2.	Resources are assigned nos. 1-9999 ONLY	change the No.
3.	There is no "TO MAKE" verb detected in last command	syntax error change statement
4.	Numeric values are not allowed for POOL & DISTRIBUTE	change statement
5.	The MAX-count of resources per data statement is 250	divide statement into two.
6.	NO command directive specified before this line	syntax error change statement
7.	This command requires a PROCESS declaration	syntax error change statement
8.	Only LIMITS or BOUNDS should have two(2) values	syntax error change statement
9.	The lower LIMIT value should be <LE> the high LIMIT	change the No.
10.	Only PRICES and COST should carry negative values	change the No.
11.	EXTRAneous data value and is not associated with a resource	syntax error change statement
12.	The relational operator is invalid i.e. <LE>, <EQ>, <GE>	syntax error change statement
13.	The (#) and (@) should always be followed by a number	syntax error change statement
14.	The incoming data DID NOT corresponds to the TABLE	count data in TABLE
15.	The table is not properly SETUP using TABLE command	syntax error change statement
16.	The INCLUDE/COPY file indicated does not EXIST!	syntax error changestatement
17.	The command directive is ILLEGAL or not in the LIST	syntax error change statement
18.	There should be ONLY one(1) relational operator	syntax error change statement

SELECTION 4. SAVE SOLUTION into solution database
option: DIF FILE creation

The previously optimized solution is saved into the solution database for reporting. The planner has the option of not saving the solution if additional changes have to be made to the analysis. Once the solution has been saved into the database, reporting through the templates are now possible.

This operation (SAVE Solution) may be executed right following the optimization when the solution is OPTIMAL. This option is requested via the UTILITY procedure.

BEST-PLAN remembers the last optimization date and time, which is normally included in the MAIN MENU. (see. sample MAIN MENU).

The planner may also request an automatic creation of a DIF file for interface to VISICALC, LOTUS 1-2-3, etc.. (see. DIFile section). Again this option is requested via the UTILITY procedure.

SELECTION 5. CHANGE SCENARIO, another analysis

This allows the planner to switch to another analysis. BEST-PLAN will request the new scenario no. desired. At this time the planner is requested if the title for the scenario is to be changed. The title is a 60-character informational title the planner assigns to a scenario analysis. Optimization is always performed on the current scenario.

The MENU template.00 may be created by the planner to summarize the scenario analysis in relation to the master PLAN. This MENU display is automatically presented to the planner on a scenario change or after saving the optimized solution. This MENU display is like a quick-look report for the planner to get KEY information needed to make planning decisions.

SELECTION 6. MAKE this SCENARIO the Master PLAN

The planner may replace the current master PLAN with the current scenario. This allows any changes to be made permanent and installed as the new PLAN. All solutions in the solution database will be deleted, except for the current scenario which will be the NEW SCENARIO1.

All files are correspondingly updated. The planner is allowed to change the title for this NEW Master PLAN.

Once the scenario has been installed as the NEW Master PLAN, SELECTION 3 is not allowed on the NEW PLAN, because the INPUT data will not be valid at this point. Re-optimization may be possible by creating another scenario (e.g. scenario 2) with or without any changes.

SELECTION 7. EDIT a REPORT Template

Data entry is accomplished using the supplied screen EDITOR. The Editor has a repertoire of control statements, called directives by which means the user (planner) can invoke a wide range of editing functions.

Since everybody have their own favorite screen editor, the system also allows the planner to link his favorite screen editor using the procedure in the installation section. (see. EDITOR section)

SELECTION 8. EXIT to PC-DOS

Done with the analysis, BEST-PLAN returns to the operating system. The system should always be terminated through this procedure to insure a proper closing of all datafiles.

SELECTION 9. Perform a (DOS) SHELL Command

This allows the planner to activate a foreign procedure from BEST-PLAN and return. The procedure could be the name of the PCDOS "batch" file. At the completion of the foreign procedure the system will reactivate BEST-PLAN with all the screen displays remaining intact.

SELECTION 10. UTILITY PROGRAM

A Utility program is supplied with BEST-PLAN to control the operation of the system. The utility program comes with its own MENU as shown:

```
╒═══════ UtilityMENU ═════════╕
│                             │
│ 1. Screen SETUP/colors      │
│ 2. FILEs Disk PATHnames     │
│ 3. OPTIMIZATION Parameters  │
│ 4. INPUT Data SYNTAX        │
│ 5. INPUT ERROR Messages     │
│ 6. Screen EDITOR Utility    │
│                             │
│ 8. Return to Main MENU      │
│ 9. DUMP/RESTORE Database    │
│ or Single-step|OPTIMIZATION │
╘═══════ KEYs(Alt)-No. ═══════╛
```

The UTILITY Section will further explain the use and intent of the UTILITY procedure.

POSSIBLE SYSTEM ERROR:

ERR= 5	BAD Database KEY, Restore the database
ERR= 6	MATH Overflow, BAD co-processor etc..
ERR= 7	Outof memory,problem too big for version
ERR=11	Division by zero, should not happen
ERR=14	Out-of-string space, out of memory ERROR
ERR=24	Device timeout ERROR, maybe printer offline
ERR=51	Internal ERROR, something went wrong!
ERR=57	Device I/O ERROR
ERR=61	Disk FULL, the RAM disk may be overloaded
ERR=70	Permission denied, file is protected
ERR=71	Disk not ready
ERR=72	Disk media ERROR, Good Luck with this one!
ERR=76	Path/File access ERROR, i.e. pathnames for DATA

POSSIBLE MATRIX ERROR:

Internal ERROR, this should not happen, trust me!

Internal ERROR, MATRIX Equation is MALFORMED

Problem too LARGE, space for NAMEs overflowed

Problem too LARGE, space for MATRIX overflowed

Problem too LARGE, space for NUMBERs overflowed

Equation SETUP, MASTER Plan not optimized

UTILITY OPERATION

A response of (enter) to all the questions will not change the currently assigned value. Please be careful in answering the questions. ONLY a Limited error checking has been implemented. The questions will recycle to confirm any changes made to the original settings.

SUMMARY OF UTILITY SELECTIONS

1. Screen Operation/Color Selections. Allows the user to change screen color/attribute parameters, plus other options.

2. Disk PATHnames for DATA/SYSTEM files. Allows the user to change PATHnames for DATA/SYSTEM files. The planner can setup his/her own data structure using subdirectories.

3. OPTIMIZATION Constants and OPTIONS. Allows the user to change optimization parameters, and reporting controls, and DIF files input/output options.

4. INPUT Data COMMAND SYNTAX definitions. Allows the user to change data entry COMMAND SYNTAX. i.e. USE/MAKE can be changed to CONSUME/PRODUCE.

5. INPUT Data ERROR MESSAGE definitions. Allows the user to change the ERROR messages presented during data entry errors.

6. Screen EDITOR/Data entry Utility. Allows the user to execute the supplied Screen EDITOR to edit files in the system.

8. EXIT to the main MENU operation. DONE with the UTILITY.

9. DUMP/RESTORE Database to/from a FILE. Allows the user to save the database to a DUMP file for later restoration (RESTORE).

or9. Single-step|OPTIMIZATION..for testing. Allows the user to single-STEP through the Optimization when debugging a MODEL. This selection replaces the above DUMP/RESTORE option when the UTILITY menu is entered through the TEST.BAT command file.

SELECTION 1. Screen Operation/Color Selections.

The user is allowed to :

1. set MONITOR mode. The monitor may be a monochrome, standard color, or a fast color adapter which does not require a sync with vertical retrace to avoid "snowing" on direct write to the frame buffer.

2. set the printer emphasize code. This is usually preceded by the ESC code. Report fields will be emphasized when printed.

3. change color assignments for all SCREENs 1-8. The planner can change the foreground and background attributes for both normal and highlighted characters. This also applies to Monochrome SCREENs. Highlighted fields are:

   report fields (bounded by the | character) and
   display highlights (bounded by (Ctl)-H character).

4. change the banner colors for LINE.1 and LINE.25 of SCREEN 1, and LINE.25 of all the remaining SCREENs.

The color code is as follows: (standard IBM color code)

0	black	4	red	8	Gray	12	light red
1	blue	5	magenta	9	light blue	13	light magenta
2	green	6	brown	10	light green	14	yellow
3	cyan	7	white	11	light cyan	15	bright white

```
     Background  Music  (1=On, 0=Off): 0(W.Tell's Overtune)
          Monitor: (M)ono,(C)olor,(F)astcolor: F
             Printer emphasize code, ESC-(xx): 69
          Printer de-emphasize code, ESC-(xx): 70

    ...enter Attributes IRGB's (FG,BG) for the following:
                                    the banner = 15, 7
  Screen 1: normal= 14, 1        HI-lite= 15, 1
  Screen 2: normal= 14, 2        HI-lite= 15, 2
  Screen 3: normal= 14, 3        HI-lite= 15, 3
  Screen 4: normal= 14, 4        HI-lite= 15, 4
  Screen 5: normal= 14, 5        HI-lite= 15, 5
  Screen 6: normal= 14, 6        HI-lite= 15, 6
  Screen 7: normal= 14, 0        HI-lite= 15, 0
  Screen 8: normal= 14, 1        HI-lite= 15, 1
```

SELECTION 2. Disk PATHnames for DATA/SYSTEM files.

The user is allowed to change the position of datafiles within his multiple drive IBM-PC system. The following can be reassigned a different than normal disk and directory assignments, all must be terminated with the backslash(\):

```
       Matrix tempFILES : 1   (1=RAMdisk,2=w/DATAfiles)
      Master PLAN FILES : DATA\
         SCENARIO FILES : DATA\
     DIF InterfaceFILES : DATA\
         Tutorial FILES : HELP\
```

TEMPORARY FILES: Temporary files are produced throughout the execution of the system as program communicate with one another. BEST-PLAN is a software system composed of fifteen (15) independent programs to make LP analysis as non- technical as possible for the planner. SCREEN displays are maintained as temporary files. Temporary files are created by the system in a RAMdisk (i.e.VDISK).

MATRIX tempFILES: These files are temporary files used to maintain the temporary DATA during the optimization procedure. These files are maintained in the same RAMdisk. For large models with limited amount of memory, these files can be maintained in a HARD disk, together with the DATA files. There will be a compromise in OPTIMIZATION speed.

MASTER PLAN FILES: Master PLAN files and the solution database are usually voluminous and will be appropriate for hard-disk drive if available.

SCENARIO FILES: Scenario files are updates to the master PLAN. They are normally small in size. Portability is appropriate for scenario files, the floppy disk will be an excellent storage of scenarios for a particular study.

DIF InterfaceFILES: These are the interfacing files for LOTUS 123, VISICALC etc. in DIF (Software Arts) format which may be created to transfer DATA to/from the mentioned software packages. The residence of these files usually depends on other applications. (see. DIFile section)

TUTORIAL FILES: All tutorial files are supplied within a sub-directory HELP. The user is allowed to relocate these files.(see. UTILITY, On-line TUTORIAL facility)

SELECTION 3. OPTIMIZATION Constants and OPTIONS.

The user is allowed to change some controllable optimization constants and reporting defaults. These specifications are preserved for the particular PLAN and its scenarios. When multiple PLANS are kept on the same disk drive, they will all carry their own unique optimization constants and OPTIONS.
They are as follows:

```
        Value to ignore in Reports:  .0001
        Value of ZERO (tolerance) :  .00001 or D.P.
                 for double precision use D.P.
             Value for timefactor :  0.075
      (use to estimate time-to-finish during optimization)
```

The above constants are used during the optimization common to most revise simplex algorithm which is used in this version of BEST-PLAN. The value of ZERO should be around: single precision =(1.E-6) or double precision =(1.E-14). The timefactor is a scaling number to approximate the time-to-finish from the product between the ROWS * COLUMNS. This may be a trial-and-error proposition based on the application.

```
            ..optimization OPTIONS:
             Minimize or Maximize : "max:"
                Objective Function: "profit"
              Optimization METHOD : primal or dual
            OPTMZ. report control : 28

              Iteration TRACE cnt.: 1
             TRACE message control: 0
```

" max: profit" is the default name for the objective function created by the system from the PRICE/COST data statements in the INPUT file. This allows the planner to change the name for a unique application. The MODEL or plan may contain any number of objective functions. The MINIMIZE and MAXIMIZE data statements allows the planner to specify these objective functions. The planner is nevertheless requested to specify which objective function is to be performed prior to optimization. You can choose between primal or dual algorithm for the solutions. You may specify to use double precision arithmetic during the optimization by using "D.P." as the value for ZERO (tolerance).

OPTIMIZATION report control
The bit-mask is as follows:

8	4	2	1

1 Enables Propagation of cost throughout the PLAN from the expenses to the product. Product margins will then be determined.

Expanding the solution. In using the relational data statements mentioned in the MODELING section, some activities (@ppp #nnnn) are not included in the MATRIX to minimize the total number of variables. These activities are re-computed by this procedure for complete reporting during analysis. This operation is automatically done when detected.

2 Enables automatic save of the solution to the DATABASE ONLY when the solution is OPTIMAL. The option is normally OFF, and the system returns to the main MENU.

4 Enables the display of the MATRIX list on SCREEN 4. The display is normally OFF unless this bit is set to 1. The current SCREEN display will be lost.

8 Enables the display of the SOLUTION list on SCREEN 3. The display is automatically enabled if there are infeasibilities in the solution. On OPTIMAL solution , the "RAW" solution list is not normally displayed.

The display of the parsed INPUTdata ErrorLIST on SCREEN 1 is automatically enabled if there are INPUT data errors.

SAMPLE: To enable the SOLUTION LIST and propagation of cost.

```
OPTMZ. report control = 8 + 1
                      = 9
```

Iteration TRACE count and the TRACE message control

The iteration TRACE count will be the number iterations between messages for tracing purposes. The proper TRACE messages are controlled by the TRACE message control. The messages are normally OFF:

The bit-mask is as follows:

4	2	1

1 reports on variables currently in the BASIS table, and variable entering/variable leaving the table during the iteration.

2 reports on the current PRIMAL non-zero variables and also the current DUAL non-zero variables.

4 reports on the current objective function value and current optimization phase. This message is automatic if any of the above have been enabled.

SAMPLE DISPLAY

```
Phase= 1  iteration= 6    current obj. value= 1133

 ...........  Primal non-zero variables:
|    #10           VL= 1E-20      |    #100          VL= 900        |
|    #20           VL= 1E-20      |    #410          VL= 1E-20      |
|    #411          VL= 40         |    #430          VL= 1E-20      |
|    #431          VL= 50         |    #460          VL= 1E-20      |
|    #461          VL= 50         |    #610          VL= 1E-20      |
|    #611          VL= 15         |    #630          VL= 1E-20      |
|    #631          VL= 15         |    #660          VL= 1E-20      |
|    #661          VL= 12         |    #710          VL= 33         |
|    #730          VL= 33         |    #760          VL= 24         |
|    #600          VL= 45         | BS a.demand#100  VL= 900        |
| BS a.e@15 #410   VL= 40         | BS a.e@15 #430   VL= 50         |
| BS a.e@15 #460   VL= 50         | BS a.e@25 #610   VL= 18         |
| BS a.e@25 #630   VL= 18         | BS a.e@25 #660   VL= 12         |
| BS a.e@25 #600   VL= 45         |
```

SAMPLE DISPLAY

```
 ..........    Dual non-zero variables:
 |     demand#310     D$= 2         |    demand#330     D$= 2          |
 |     demand#360     D$= 2         |    supply#510     D$= 2          |


 ..........   <#510> entered basis and <a.supply#510> left basis

 ..........   Current basic variables are:
 |a.demand#10  |a.demand#100 |a.demand#11  |a.demand#20  |a.demand#21  |
 |#310         |#330         |#360         |a.e@15 #410  |a.e@15 #430  |
 |a.e@15 #460  |a.e@25 #610  |a.e@25 #630  |a.e@25 #660  |a.supply#210 |
 |a.supply#230 |a.supply#260 |a.supply#51  |#510         |a.supply#52  |
 |a.supply#53  |a.supply#530 |a.supply#560 |a.e@25 #600  |
```

Other considerations:

ARTIFICIAL variables — will carry the prefix "a." with the associating ROW name.

SLACK variables — will carry the prefix "s." with the associating ROW name.

BEST-PLAN formulates the "e" ROWS for POOL, EQUATION and COMBINE data statements. It also formulates the "supply" and "demand" ROWS for fix-yield relational data statements.

DIF File INPUT-OUTPUT Option

This will setup the environment to accept a DIF INPUT for data entry on a scenario analysis and also to automatically create a DIF file for input to a SPREADSHEET system. (see. DIFile section) This will then allow full interface between a VISICALC, LOTUS DEV. 1-2-3 system and the BEST-PLAN optimization system.

```
            DIF input-file option: No
            DIF outputfile option: No

      ....if DIF outputfile option: Yes
        Enter the selected data and order (e.g.*,A#,D,G,B,C).......
         ....DIF Interface OPTIONS: *,A#,D,B,C
               *=put data title in first record
 A#=resources as labels=@ppp #nnnn A=resources as nos.=pppnnnn
  B=name.id(30-chrs) C=volume      D=prices         E=int.cost
  F=total.$$         G=incentive   H=low.lmt        I=high.lmt
```

The DIF file(INPUT/OUTPUT) will be assigned the following filenames:

```
INPUT  FILENAME: PnnnnDss.DIF
OUTPUT FILENAME: PnnnnSss.DIF
                 nnnn = PLAN no.
                   ss = scenario
```

PnnnnDss.DIF may be created by the SPREADSHEET system using the corresponding SAVE command with column-wise orientation.

PnnnnSss.DIF is created by BEST-PLAN during the SAVE Solution procedure (i.e. selection 10).

SELECTION 4. INPUT Data COMMAND SYNTAX definitions.

The user is allowed to create his own language for data entry. The various keywords used by BEST-PLAN during INPUT data analysis are kept in a table which could be modified for certain application. Be careful against redundancies. Foreign Languages could be implemented by replacing the data entry keywords with their equivalents. The equal-sign (=) is used as delimiters.

The DATA SYNTAX table is as follows: ALL WORDS IN A LINE ARE SYNONYMOUS and can used interchangeably. Limit each line to 64 characters.

1. =INCLUDE=COPY
2. =MACHINE=PROCESS=SECTION=PLANT=UNIT
3. =RESOURCE=VARIABLES=MATERIALS=ACTIVITY
4. =MAXIMIZE=MINIMIZE

(parameter data statements)

5. =PRICES=VALUES
6. =COSTS=EXPENSES
7. =MINIMUM=LOW=L.L.=LOWERB=LOW.LMT
8. =MAXIMUM=HIGH=U.L.=UPPERB=HIGH.LMT
9. =FIXED=HOLD=SET
10. =LIMITS=BOUNDS=RANGE

(equation or relational data statements)

11. =USE=SPLIT=COMBINE=ACCUMULATE=INPUTS
12. =POOL=INVENTORY=WIP
13. =DISTRIBUTE=SUPPLY
14. =APPLYCOST=ADDCOST
15. =EQUATIONS=EQUATE=CONSTRAINTS=ROWS
16. =XFERCOST=XFER$
17. =TABLE=TABULATION
18. =DATA
19. =END=ENDDATA

(removing or deleting relational data statements)

20. =REMOVE=DROP=DELETE

(command preposition for relational data statements)

21. =FROM=WITH=FOR=TO=MAKE=TOMAKE=INTO=OUTPUTS

(relational operators for equations)

22. =<LE>=<LT>
23. =<EQ>=EQUALS
24. =<GE>=<GT>
25. =<RN>=<RANGE>

SELECTION 5. INPUT Data ERROR MESSAGE definitions.

The user is allowed to create his own language for data entry. The various error messages used by BEST-PLAN during INPUT data analysis are kept in a table which could be modified for certain application. This convenience is intended for foreign languages to have the necessary equivalent error messages on data entry errors. Limit each line to 64 characters.

The ERROR MESSAGES table is as follows:

1. Processes are assigned nos. 1-999 ONLY
2. Resources are assigned nos. 1-9999 ONLY

3. There is no "TO MAKE" verb detected in last command
4. Numeric values are not allowed for POOL & DISTRIBUTE
5. The MAX-count of resources per data statement is 250
6. NO command directive specified before this line

7. This command requires a PROCESS declaration
8. Only LIMITS or BOUNDS should carrytwo(2) values
9. The lower LIMIT value should be <LE> the high LIMIT

10. Only PRICES and COST should carry negative values
11. EXTRAneous data value and is not associated with a resource

12. The relational operator is invalid, must be one of these <LE>,<EQ>,<GE> or <RN>

13. The (#) and (@) should always be followed by a number
14. The incoming data DID NOT corresponds to the TABLE
15. The table is not properly SETUP using TABLE

16. The INCLUDE/COPY file indicated does not EXIST!
17. The command directive is ILLEGAL or not in the LIST
18. There should be ONLY one(1) relational operator
19. ...not used
20. ...not used

SELECTION 6. Screen EDITOR/Data entry Utility.

The user is allowed the stand-alone use of the supplied Screen EDITOR for modifying the input DATA during MODEL debugging. The Editor has a repertoire of control keys, called directives by means of which the user (planner) can invoke a wide range of editing functions. Since everybody have their own favorite screen editor, the system also allows the planner to link his favorite screen editor using the procedure in the installation section. (see. EDITOR section)

SELECTION 8. EXIT to the main MENU operations.

Done with theUTILITY. BEST-PLAN returns to the main MENU operation.

SELECTION 9. DUMP/RESTORE Database to/from a FILE.

During the course of system usage for a given application, the size of the DATABASE may have to be increased or decreased. A NEW DATABASE file is to be created which will have a lower or higher record capacity. The contents of the ORIGINAL DATABASE will have to be DUMPed into a file and RESTOREd into the newly created DATABASE. Also, it is a sound practice to keep a backup of the SOLUTION database.

The program will create a dump file containing the contents of the current DATABASE, in the proper sorted order. This file can be saved by the user for future restoration. The user may then RE-INITIALIZE the DATABASE file. A NEW MAXIMUM record count may be specified and a corresponding RESTORE will restore the data into the newly created DATABASE.

SELECTION 9. Single-step|OPTIMIZATION..for testing.

The user is allowed to single-STEP through the Optimization when debugging a MODEL. This selection replaces the above DUMP/RESTORE option when the UTILITY menu is entered through the TEST.BAT command file.

The main OPTIMIZATION procedure is divided into four(4) passes. They are as follows:

PASS 1: The Input DATA is read and the statements SYNTAX are properly analyzed. Any possible ERROR messages are presented with a copy of the input DATA, as parsed by the PASS 1 compiler. SCREEN 2 is used to display these messages. A temporary intermediate file is created for use by the next PASS. If there are NO ERROR messages, Screen 2 will display the scenario Input DATA and Screen 3 will display the Master PLAN Input DATA.

PASS 2: The intermediate file is read and the MATRIX equations are formulated. For Scenario UPDATES, the master PLAN MATRIX is modified with the changes for a new MATRIX. SCREEN 4 is used to display the new MATRIX.

PASS 3: The new MATRIX is optimized using a REVISED-SIMPLEX algorithm. TRACE messages may be requested by setting the necessary option (see SELECTION 3.) SCREEN 3 is used to display these messages.

PASS 4: A "raw" SOLUTION list is created for display, also using SCREEN 3. Any possible solution expansion is handled by PASS 4, as well as the internal COST propagation COMPUTATIONs as mentioned in the Planning MANUAL. (i.e. APPLY, XFERCOST)

Using this option, the user can Single-STEP through the above mentioned PASSes. This will allow a faster MODEL debugging process. The function-key F10 may be used to ABORT the OPTIMIZATION at any stage. The combination of SELECTION 6 (Screen EDITOR) and SELECTION 9 (Single-STEP: OPTIMIZATION) can speed up MODEL debugging tremendously.

OTHER CONSIDERATIONS

Installing your favorite SCREEN Editor and/or word processor for DATA ENTRY/UPDATE. BEST-PLAN allows automatic invocation of a screen editor or word processor to accomplish data entry and/or update of the templates. This is accomplished by creating a standard PC-DOS "batch" file which is activated by BEST-PLAN upon request.

The supplied SAMPLE "batch" file for use with Wordstar is as shown: BEST-PLAN will supply the necessary parameters.

```
FILENAME:  MODIFY.BAT
         1.    echo off
         2.    cd %2
         3.    ws %1
         4.    cd \bp.dir
         5.    bp-sys%3
         6.    continue
```

Parameters are:

```
%1 = filename to EDIT
%2 = disk/directory where FILE resides
%3 = switch for calling program
```

A change directory command is included to return to BEST-PLAN's directory upon completion.

The supplied BP.BAT for use to start-up the BEST-PLAN system as shown: The main PROGRAM is called BP-SYS.

```
FILENAME:  BP.BAT
      1.    echo off
         2.    bp-sys %1
         3.    continue
```

The supplied TEST.BAT for use to start-up the BEST-PLAN system as shown:

```
FILENAME:  TEST.BAT
      1.    echo off
         2.    bp-sys/0
     3.    continue
```

DEFAULT MAIN MENU: The following is supplied as the default main MENU if Report Template.00 is not yet available for a given application: This file must be resident in the same disk/directory as the master PLAN. When both Template.00 and BP-DEF.F00 do not exist, the system will use the INPUT data Pnnnn-ss.IN as Template.00. This allows the user to create an INPUT file which also works as the REPORT file to superimpose input and solution. The user must be responsible for managing such combination using input comment lines.

FILENAME: BP-DEF.F00

```
╔══════════════════════════════════════════════════════════════════╗
║            BEST-PLAN   Profit Optimization System                ║
║ |title                                                        |  ║
╠═══════════════════════════╦══════════════════════════════════════╣
║ plan:|plan| scenario:|sn| ║  last analysis:  |sndate| |sntime|   ║
╟───────────────────────────╢  cost(minimized) |optimal=t$.e42 |   ║
║ selections:               ║ infeasibilities  |inf=vl|            ║
║ 1. report writing         ╟┬────────────────────────────────────┬╢
║ 2. update/data entry      ║├────────────────────────────────────┤║
║ 3. ..OPTIMIZATION...      ║│ This is the default TEMPLATE.00 for│║
║ 4. save NEW solution      ║│ BEST-PLAN. To REPLACE, create FILE:│║
║ 5. change/remove scenario ║│         Pnnnn.F00                  │║
║ 6. make scenario=>MASTER  ║│             where  nnnn=PLAN NO.   │║
║ 7. edit template(REPORT)  ║│                                    │║
║ 8. EXIT to IBM-PC D O S   ║│ The TEMPLATE serves as a QUICKLOOK │║
║ 9. special INTERFACES     ║│ display of the last optimization's │║
╟───────────────────────────╢│ SOLUTION in the DATABASE.          │║
║   The ROSETTA SYSTEMS     ║│ (ref. Report templates.)           │║
║    decision  support      ║├────────────────────────────────────┤║
╚═══════════════════════════╩╧════════════════════════════════════╧╝
```

APPLICATIONs (MODELS) DISPATCHING FACILITY

BEST-PLAN allows the planner to setup an APPLICATION dispatching MENU, just like the DEMO SYSTEM and the supplied six(6) sample applications. This was accomplished by creating a MENU file with the filename "BP-TBL" using any screen editor preceded by:

```
PLANS=nnnn/nnnn/nnnn/nnnn/nnnn/nnnn/nnnn/nnnn/nnnn
```

Here the position of the PLAN number nnnn corresponds to the (ALT)-number selected. e.g

```
PLANS=1901/1902/1903/1982/1983/1984/nnnn/nnnn/nnnn
        1    2    3    4    5    6    7    8    9
```

While under the "BP-TBL" condition the user can select PLAN:1982 by (ALT)-4, or PLAN: 1902 by (ALT)-2. The application dispatcher initializes with the last remembered session. The "BP-TBL" file must be resident in the same disk/directory as the master PLANs. To highlight words or group of words, (Ctl)-H are added to the text. The supplied "BP-TBL" for the DEMO SYSTEM is as follows:

```
PLANS=1901/1902/1903/1982/1983/1984

╔══════════════════════════════════════════════════════╗
║          BEST-PLAN  SAMPLE DEMONSTRATION SYSTEM      ║
╠══════════════════════════════════════════════════════╣
║                                                      ║
║ ..Mathematical Formulation Examples                  ║
║  1.       PLAN: 1901   small Distribution Model      ║
║  2.       PLAN: 1902   Animal Feed (Diet) Model      ║
║  3.       PLAN: 1903   Multi-period scheduling       ║
║ ..STEP-by-STEP Flow Representation Examples          ║
║  4.    PLAN: 1982   Raw Material Allocation Model    ║
║  5.    PLAN: 1983   Energy Planning System           ║
║  6.    PLAN: 1984   Production Planning Model        ║
║             ...............................          ║
║     Up to nine(9) PLANs may be supported by this     ║
║     application DISPATCHER capability.  (BP-TBL)     ║
║             ...............................          ║
║  0.      .......EXIT back to  P C  D O S.......      ║
║                                                      ║
╚═══════════════ PRESS (alt) and the Number ═══════════╝

enable tutorial= (alt)-H
```

ON-LINE TUTORIAL FACILITY

While in the "BP-TBL", main MENU, UTILITY and Screen EDITOR procedure, the ON-LINE TUTORIAL facility may be activated by pressing (ALT)-H. This facility is only applicable when the TUTORIAL files are properly installed using the DEMO Diskette supplied with the system. The DEMO Diskette contains a sub-directory HELP which contains all the files for the Tutorial. The TUTORIAL information is specified through a TABLE.

The TUTORIAL TABLES, #SYSM, #MENU, #UTLT, and #EDIT are supplied to specify the tutorial to be displayed when a corresponding (Alt)-"n" is pressed. The format is as follows:

```
ALT,SS,"filename"

where:  ALT = the number "n" in (Alt)"n"
        SS  = the SCREEN number to contain display. (1-4)
   filename = the DOS filename
```

These TUTORIAL files are straight ASCII files which may be modified by the planner for a given application. THEREFORE, for certain applications, the planner may develop his own TUTORIAL FILES to facilitate system use and operations (a custom-tailored HELP) for non-technical users. To highlight words or group of words, (Ctl)-H can be added to the text. The supplied SCREEN EDITOR has been adjusted to properly edit the TUTORIAL files by processing them the way they are presented. Within the SCREEN EDITOR, the (Ctl)-H character is used to ENABLE/DISABLE highlights.

Not all TUTORIAL files are needed at the same time. Due to storage limitations, some of the TUTORIAL may be eliminated. #BLANK has been created to indicate missing TUTORIAL. Nevertheless, it is a good practice to eliminate TUTORIAL in groups. (i.e all DEMO application TUTORIAL files are in #SYSM*.*)

In summary, (ALT)-H will ENABLE/DISABLE the TUTORIAL facility.

BEST-PLAN S C R E E N E D I T O R

The Screen EDITOR is similar to WORDSTAR, most of its control keys are the same and its approach to handling the screen is about the same. Instead of the two (2) control characters, (Alt) is used with the last character in most commands, i.e. (Ctl)-KY for block delete is changed to (Alt)-Y. The SCREEN EDITOR supplied is not intended for large data creation. There is a limitation of 400 lines and all lines cannot be longer than 80 characters. (i.e. BEST-PLAN is screen oriented.) All lines are truncated before it is SAVED. You can still manipulate lines >80 characters while in the BUFFER. The user can still use his favorite EDITOR, e.g. WORDSTAR for the larger data creation. Nevertheless, "WHAT-IF" analysis are readily handled, since "WHAT-IF" update-FILEs contain ONLY the changes to the MASTER PLAN.

SAVING the edited FILE

(Alt)-S
: Save the file to disk. No BACKUPS are made.

(Alt)-E
(Alt)-Q
: Exit or Quit the Editor, the user is requested to save the file if changes were made.

(Alt)-R
: Reads another FILE before the current line position. The name of the FILE will be requested. The user must specify the full PATHname of the FILE. Special FILE tags are available to speed typing, i.e.:
    1. In editing DATA files, "sNN" means the data for scenario NN.
    2. In editing TEMPLATEs, "tNN" means the template NN.

(Alt)-P
: Quick print of the buffer. The contents of the buffer is transferred to the line printer. (LPT1:) This may also pertain to any SCREEN(1-8) display. Use F10(ABORT) to stop printing when necessary.

(Alt)-Z (ZOOMING-IN or OUT)
: Allows the enlargement or reduction of the editing window. This is disabled when editing TUTORIAL files.

Screen EDITOR KEYS:

The following list defines the meaning of each key.

Cursor keys (arrows on numeric keypad)
: Move the cursor up, down, left, or right one position. If the cursor would move off of the screen in a direction in which there is more data, then scroll the screen in that direction 1/4 the current window size.

Ins
: Complement the INSERT mode switch. INSERT may be enabled or disabled as the case may be. This key is also used to disable LINE drawing.

Character keys
: Enter the character into the current line according to current INSERT mode. (Ctl)-P or DLE, is used to precede CTL characters, i.e. (Ctl-H). The ALT can also be used to enter extended characters.

TAB
: TABs are emulated using modulo 8 for tab positions, and are maintained as spaces rather than the TAB character. TABs can be inserted by the EDITOR in the text file using the (Ctl)-P or DLE preceding the TAB.

Backspace
: If INSERT ON, backspace over and delete the character. If INSERT OFF, backspace over and replace with a space.

Del
: If INSERT ON, delete the character above the cursor. If INSERT OFF, replace with a space, then move to the right.

End
: Position the cursor at the start of the last line on the screen.

Home
: Position the cursor at the start of the first line on the screen.

Page down
: Scroll the screen 22 lines down for the next page.

Page up
: Scroll the screen 22 lines up for the next page.

Enter key
: If INSERT ON, create another line. If INSERT off, position the cursor at the start of the next line.

SPECIAL cursor COMMANDS for DRAWING LINES

(Alt)-minus
Set LINE character to single line. Line drawing always goes in overlay mode, and not inserted.

(Alt)-equal
Set LINE character to double line. Line drawing always goes in overlay mode, and not inserted.

(Atl)-zero
Set LINE character for erase line. This done by retracing it using appropriate DRAW line commands. The "space" character can also be used to erase small line segments or "overshoots". The Ins (INSERT) key DISABLES LINE drawing.

(Ctl)-right cursor
Propagate the LINE character (if specified) or last character typed towards the right.

(Ctl)-left cursor
Propagate the LINE character (if specified) or last character typed towards the left.

(Ctl)-down cursor
Propagate the LINE character (if specified) or last character typed towards the bottom.

(Ctl)-up cursor
Propagate the LINE character (if specified) or last character typed towards the top.

SPECIAL COMMANDS for TEXT operations

(Alt)-J
Join the next line onto the current one at the end. ON OUTPUT, all lines are truncated to 80-characters long, so care must be exercised in joining lines.

(Alt)-G
Split the current line into two parts at the cursor position. At column one this adds a new lines before the current one. This works similar to the ENTER KEY when INSERT is ON.

(Alt)-K
Erase the rest of the line starting at the current cursor position. If current position equals first column position, then erase the whole line.

(Alt)-X
If INSERT ON, delete characters before the current cursor position. If INSERT OFF, replace all with spaces.

(Alt)-I
Add a line before the current line position.

(Alt)-D
Delete the current line.

(Alt)-O
Move to end of line, or after last non-space position.

(Alt)-U
Move to start of word, or the next space position. If end of line is reached, use the next line.

(Ctl)-end
Move to the absolute END of the file.

(Ctl)-home
Move to the absolute START of the file.

SPECIAL COMMANDS for BLOCK operations

(Alt)-V
Set BLOCK Orientation. A BLOCK can be horizontally oriented (REGULAR) by lines, or a vertical BOX.

(Alt)-B
Mark the current line as the start of BLOCK. The BLOCK is displayed in highlighted mode.

(Alt)-N
Terminates the BLOCK. A second (Alt)-N may be used to enlarge or reduce the BLOCK as the case maybe. Positioning the end of BLOCK above the start of BLOCK will delete the BLOCK.

The following operations are affected by the current INSERT mode, when BOX BLOCKs are specified. If INSERT ON, COPY and MOVE will insert the block. If INSERT OFF, the block is OVERLAYED in the new place. In COPY, the old position is replaced with spaces.

(Alt)-Y
Delete a marked lines or a BLOCK, see (Alt)-B, how to create a BLOCK.

(Alt)-C
Copies a marked lines before the current line or the BOX BLOCK at cursor position. The copied BLOCK will be the new BLOCK. see (Alt)-B, how to create a BLOCK.

(Alt)-M

Moves a marked lines before the current line or the BOX BLOCK at cursor position. see (Alt)-B, how to create a BLOCK.

(Alt)-W
Writes the marked lines to another FILE. This does not apply to BOX BLOCK. The name of the FILE will be requested. Enter the full PATHname of the file as the case may be.

SPECIAL COMMANDS for PATTERN SEARCH/REPLACE operations

(Alt)-F
Start search for a particular pattern. The string pattern will be requested.

(Alt)-A
Start search for a particular pattern and replace it with the replacement string. The string pattern will be requested, as well as the replacement string. The editor reconfirms whether the change is to be made or not.

(Alt)-L
Continue the pattern search as started by the previous (Alt)-F or (Alt)-A instructions.

TUTORIAL WHILE USING THE SCREEN EDITOR

TUTORIAL is available while using the EDITOR. The TUTORIAL is activated by the (Alt)-H key. The Screens 2, 3 and 4 will carry the TUTORIAL. The window containing the editor (Screen 1.) will be suspended while the TUTORIAL is active. AGAIN, these same TUTORIAL can be edited by the EDITOR to suit a particular application.

The following SUBJECTS are displayed by using the (Alt)-"n", wherein "n" is the number 0-9.

(Alt)-0
The main SCREEN EDITOR command KEYS instructions will be displayed in Screens 2, 3 and 4. Screen 2 will contain the TABLE of CONTENTS for the TUTORIAL messages, same as this page in this documentation.

(Alt)-1
The SAMPLE Distribution problem will be displayed in Screens 2, 3 and 4.

(Alt)-2
The SAMPLE Diet-Blend problem will be displayed in Screens 2, 3 and 4.

TUTORIAL WHILE USING THE SCREEN EDITOR (cont)

(Alt)-3
Discussions on the OPTIMIZATION run will be displayed on Screens 2, 3 and 4.

(Alt)-4
Discussions on the STEP-by-STEP Material FLOW Representation METHOD.
Screen 2: Resources, Processes and Activities.
Screen 3: Relationship between Resources and Processes.
Screen 4: PRICES/COST and LIMITS for resources.

(Alt)-5
Discussions on Relational data statements.
Screen 2: Differences between FIX and VARIABLE relational statements.
Screen 3: Special Relational data statements. POOL and DISTRIBUTE
Screen 4: Computational data statements. XFERCOST and APPLY.

(Alt)-6
Additional data statements. INCLUDE, REMOVE and TABLES.
Screen 2: INCLUDE, REMOVE and TABLE.
Screen 3: How to setup TABLEs for data.
Screen 4: SAMPLE table for the SIMPLE distribution problem.

(Alt)-7
SUMMARY for the STEP-by-STEP Material Flow representation METHOD. Discussions on SYNTAX and RULES for data statements.

(Alt)-8
Discussion on the Mathematical/EQUATIONs METHOD of modeling. Screen 4 will carry a SAMPLE diet/Blend problem.

(Alt)-9
Discussions on HOW to create Report TEMPLATES
Screens 2, 3 and 4 will carry the Report TEMPLATE section of the documentation.

BEST-PLAN R E P O R T W R I T E R

REPORT is a program which supports a specially tailored fill-in-the-blank (template technique) reporting writing. It is specially suited for this system where data information is not organized or structured into a hierarchal network. REPORT will tie together all data which are separately stored in separate records in the solution
database.

The basic idea is to create, using the standard available personal editor, a report page or pages. The report page will have the titles, narratives, column/row headings and other information as needed. The variable data fields which are to be supplied by REPORT will be identified with field identifiers. These field identifiers will identify the data to be retrieved from the database.

Data values are displayed using field replacement. Fields are identified by:

```
|dataname=datatype.modifiers|
```

SAMPLE REPORT ENTRY

```
╔══════════════════════════════════════════════════════════════════════╗
║plant #1 cost summary       oil costs      op. costs      total cost   ║
║                                                                       ║
║d10 & d30 |#100@10.t1|mbl |#100@10.d2.t2 ||#11@10.d2.t3||fn=(f2+f3).d2.t4|║
║d30 & d60 |#100@11.t1|mbl |#100@11.d2.t2 ||#11@11.d2.t3||fn=(f2+f3).d2.t4|║
║d10 & d60 |#100@12.t1|mbl |#100@12.d2.t2 ||#11@12.d2.t3||fn=(f2+f3).d2.t4|║
║                                                                       ║
╚══════════════════════════════════════════════════════════════════════╝
```

In summary, all field identifiers will be replaced by its corresponding values plus the required editing before the page is presented as a report. This report page is called a report template. The system supports up to 99 report templates per plan.

BEST-PLAN supports up to 99 report templates per PLAN. For convenience, it is better to create multiple templates of one-screen size (23-lines) rather than one long template. Here the template number will act as the page number and facilitate random access capability by page.

COST SUMMARY REPORT

The following shows a report template.

```
.df=vl.e34
.d2=t$.e80
.d3=t4/3.0

     A.B.C. Chemical COMPANY              today's date: |today |

     cost summary report    PLAN:|plan| scenario:|sn|

     demonstration:         lubricating oil plant optimization

     scenario: |title                                              |

    cost(minimized): |optimal=t$.e42   |   infeasibility: |inf=t$.e34|
      analysis date: |sndate|                     count: |inf=vl|
               time: |sntime|

 plant #1 cost summary        oil costs       op. costs       total cost

d10 & d30 |#100@10.t1|mbl |#100@10.d2.t2 ||#11@10.d2.t3||fn=(f2+f3).d2.t4|
d30 & d60 |#100@11.t1|mbl |#100@11.d2.t2 ||#11@11.d2.t3||fn=(f2+f3).d2.t4|
d10 & d60 |#100@12.t1|mbl |#100@12.d2.t2 ||#11@12.d2.t3||fn=(f2+f3).d2.t4|
--------------------------------------------------------------------------
totals    |t1        |mbl |t2            ||t3          ||t4              |

   plant #2 cost summary       plt #1 cost   plt #2 op. cost    total cost

m10 product |#510.t6 |mbl |fn=d3.d2.t7   ||#21@20.d2.t8||fn=(f2+f3).d2.t9|
m30 product |#530.t6 |mbl |fn=d3.d2.t7   ||#21@21.d2.t8||fn=(f2+f3).d2.t9|
m60 product |#560.t6 |mbl |fn=d3.d2.t7   ||#21@22.d2.t8||fn=(f2+f3).d2.t9|
--------------------------------------------------------------------------
totals      |t6      |mbl |t7            ||t8          ||t9              |
--------------------------------------------------------------------------
              total operating costs  (plant #1 & #2)    |t9              |

              products         volume           overall costs/mbbls

             m10 product    |#710.t10.e32 |        |#510=i$.e80   |
             m30 product    |#730.t10.e32 |        |#530=i$.e80   |
             m60 product    |#760.t10.e32 |        |#560=i$.e80   |
```

RULES FOR CREATING A REPORT TEMPLATE

- Incorporate all report titles, narratives, column and row headings, and other cosmetic elements for the report.

- Identify the field size of a specific data element using two (|) -characters. There must be at least one space at the beginning of a line prior to the identifier's (|) character.

```
          RESOURCES          VOLUME           PRICE

     |...data field...|   |..data field..|  |.data field.|

where: |...data field...|  =  |dataname=datatype.modifiers|
```

- Identify the dataname. There are eleven (11) possible datanames.

  dataname: The following are supported.
  (Both upper and lower case characters are supported but not combinations.)

```
      plan          = PLAN number (4-digits)
      sn            = SCENARIO number (2-digits)
      title         = the (60) chr. summary or title
      optimal       = the objective function value or name
      infeas        = the infeasibility value or total count
      sndate        = scenario optimization date
      sntime        = scenario optimization time
      today         = today's date
      fn            = arithmetic function
                  (computational expression)
      tNN           = total register, NN=register no.
                     (up to 50 registers)
      #nnnn         = resources or activities
or @ppp#nnnn   #nnnn -resource no.,@ppp-process no.
or #nnnn@ppp
```

```
      |...data field...|  =  |dataname=datatype.modifiers|
```

- Identify the datatype information needed for the field. Depending on the dataname the following shows the possible datatype information available:

```
  datatype: The following are supported.
      (Both upper and lower case characters are supported but not
      combinations.)

      for dataname = #nnnn
                           = @ppp #nnnn
                           = #nnnn @ppp

           =vl       volume or total count
           =t$       total value (volume*price)
           =ll,=ul   lower and upper limits,respectively
           =p$       unit price or cost
           =r$       reduced cost or shadow price
           =i$       internal cost (cost propagation)
           =nm       name identification

           The default is =vl(volume).

      for dataname = optimal

           =vl,=t$   value of the OBJECTIVE FUNCTION
           =nm       OBJ. FN.'s name (6-characters)

      for dataname = infeas

           =vl       number of INFEASIBILITIES
           =t$       value of total INFEASIBILITIES

      for special definition

           =dNN      NN=definition number.
                     (up to 50 definitions)
```

```
 |...data field...|  =  |dataname=datatype.modifiers|
```

- The data field may be further qualified using a field modifier. There are four (4) general types of field modifiers available:

```
modifiers:  The following are supported.
       (Both upper and lower case characters are supported but
       not combinations.)

    .sNN   scenario's solution     NN=scenario no.
    .tNN   add-to-register         NN=register no.
                                   (up to 50 registers)
    .dNN   definition(additional)  NN=definition no.
                                   (up to 50 definitions)
    .eNN   editing instruction,    NN=editing code


The defaults are as follows:

      the scenario    is always the current scenario.
      add-to-register is always OFF.
```

- By default, if the data value does not exist in the database the field is blanked out. If the data exists, the default editing formats are as follows:

```
       for volumes two  (2)  decimal  positions,  commas  every
                             thousands

       for prices  two (2) decimal positions, commas and dollar
                   sign

       for optimal no decimal position, commas and (& infeas)
                             dollar sign.
```

The following are editing specification numbers when the defaults are not desired.

The following are the supported editing codes:
(Both upper and lower case characters are supported but not combinations.)

```
a. .e1(0)-.e1(9) scientific notations     0-9 dec.pos.
   .e5(0)-.e5(9) same    except absolute value of data

b. .e2(0)-.e2(9) decimal format,no commas 0-9 dec.pos.
   .e6(0)-.e6(9) same    except absolute value of data

c. .e3(0)-.e3(9) decimal format,w/ commas 0-9 dec.pos.
   .e7(0)-.e7(9) same    except absolute value of data

d. .e4(0)-.e4(9) monetary format with $   0-9 dec.pos.
   .e8(0)-.e8(9) same    except absolute value of data
```

ADD-TO-REGISTERS "TOTAL"

A data value can be added into a "total" register for later printing. This is indicated by the field modifier .tNN (NN=register no.). A corresponding dataname tNN in another field will later indicate that the contents of the register be printed out.

```
  Products        Volume             Op. Cost

 d10 & d30   |#100@10=vl.t1| mbl |#100@10=t$.t2 |
 d30 & d60   |#100@11=vl.t1| mbl |#100@11=t$.t2 |
 d10 & d60   |#100@12=vl.t1| mbl |#100@12=t$.t2 |
     -----------------------------------------------
 totals      |t1           | mbl |t2            |
```

The editing specification of the "total" field will be the same as the editing specification of the fields to be added. BEST-PLAN remembers the editing specification of the first field added to this register.

ADDITIONAL DEFINITIONS

It should be noted that the field may be too small to contain all the field definitions/modifiers needed. Special additional field definitions may be created and specified before the first line of the report template. This extra definition may then be recalled in fields where they apply. A default definition is also supported, which is global to the entire report template.

(.df=..DEFINITIONS)

```
.df=vl.e34
.d2=t$.e80
.d3=t4/3.0

   SAMPLE REPORT

   plant #2 cost summary     plt #1 cost  plt #2 op. cost    total cost

m10 product |#510.t6  |mbl |fn=d3.d2.t7  ||#21@20.d2.t8||fn=(f2+f3).d2.t9|
m30 product |#530.t6  |mbl |fn=d3.d2.t7  ||#21@21.d2.t8||fn=(f2+f3).d2.t9|
m60 product |#560.t6  |mbl |fn=d3.d2.t7  ||#21@22.d2.t8||fn=(f2+f3).d2.t9|
    ---------------------------------------------------------------------
totals      |t6       |mbl |t7           ||t8          ||t9              |
```

Here the default definition is .vl.e34. This definition applies to all the field in the report template.

All definitions must PRECEDE the template starting at LINE.1 and all definitions must be preceded with a dot(.). These definitions are all stored into a table for future use.

Additional definition may be used to contain computational equations or functions.(see computational capability). Up to 50 definitions are available per report template.

FIELD VALUES

BEST-PLAN remembers all the field values for the current report LINE. A report LINE may have up to 50 fields. These fields may be referenced using the symbol fNN (NN=field number). Before using a field value in an arithmetic expression, it must be previously defined in a preceding data field.(i.e. an arithmetic expression in FIELD 5 can only reference fields 1-4.)

> Field values are remembered from the last occurrence of the field. Field values are cleared at the beginning of a template line. "Total" registers are remembered during the entire report.

COMPUTATIONAL CAPABILITY

REPORT supports an extensive computational capability by allowing the planner to specify equations or computations for a particular report field. This arithmetic function observes the standard hierarchal discipline, i.e. order of priority

```
() parenthesis (inside-to-outside)
^  exponentiation
* and / multiply and divide
+ and - add and subtract
```

The FORMAT is as follows:

```
      |fn=(..ARITHMETIC EXPRESSION..).modifier|

 or  |fn=dNN.modifier   |
        NN = definition number
```

If the expression is within the field, it must be enclosed within parenthesis. If additional definitions were used, the definitions must be previously specified within the definition section.

COMPUTATIONAL CAPABILITY

The arithmetic expression may contain the following elements:

```
      1. constants. i.e.  45.0, 1.0E20
      2. Field values:
         fNN     where NN= field no.
      3. "TOTAL" register:
         tNN     where NN= register no.
      4. Resource solution DATA:(within parenthesis)
           (#nnnn=XX)or (@ppp #nnnn=XX)

      where #nnnn is the resource number. Activities may be specified
      in either formats.  @ppp #nnnn or #nnnn @ppp

            =XX      is the datatype. i.e.
            =vl      volume
            =t$      total value
            =ll,=ul  lower and upper limits
            =p$      prices/costs


       |fn=(f2*3.5+((#100@10=t$)/t5))|

       where:   f2              = field value
                                  FIELD 2
                3.5             = a constant
                (#100@10=t$)    = a solution data
                t5              = totalling register
                                  REGISTER 5
```

Use parenthesis FREELY to separate the elements in the expression for good visibility.

```
.d1=(#100)/3.5*(#200)+(#300)

      |fn=d1.e43   |
      where:    d1               = definition to contain
                                   the expression
                #100,#200,#300   = resources, volume
                                   data is default
                e43              = editing specification
```

```
DEMONSTRATION PROBLEM:      A.B.C. Chemical Company
                            Produces three(3) oil based
                            products through 2 plants.
```

```
.df=vl.e34

     A.B.C. Chemical COMPANY                    todays date: |today |

     cost summary report (comparison  plan vs analyses 2,3&4)
     demonstration:         lubricating oil plant optimization

  |title.s1                                                |
  |title.s2                                                |
  |title.s3                                                |
  |title.s4                                                |

              cost(minimized)      analysis   date&time    infeasibility
  plan      : |optimal=t$.s1.e42|  |sndate.s1| |sntime.s1||inf=t$.s1.e|
  analysis 2: |optimal=t$.s2.e42|  |sndate.s2| |sntime.s2||inf=t$.s2.e|
  analysis 3: |optimal=t$.s3.e42|  |sndate.s3| |sntime.s3||inf=t$.s3.e|
  analysis 4: |optimal=t$.s4.e42|  |sndate.s4| |sntime.s4||inf=t$.s4.e|

             plant 1 production    plant 2 production  overall cost/unit

 plan       : d10 |#210.s1.e34  |  m10 |#510.s1.e34  | |#510=i$.s1.e82   |
            : d30 |#230.s1.e34  |  m30 |#530.s1.e34  | |#530=i$.s1.e82   |
            : d60 |#260.s1.e34  |  m60 |#560.s1.e34  | |#560=i$.s1.e82   |

 analysis 2 : d10 |#210.s2.e34  |  m10 |#510.s2.e34  | |#510=i$.s2.e82   |
            : d30 |#230.s2.e34  |  m30 |#530.s2.e34  | |#530=i$.s2.e82   |
            : d60 |#260.s2.e34  |  m60 |#560.s2.e34  | |#560=i$.s2.e82   |

 analysis 3 : d10 |#210.s3.e34  |  m10 |#510.s3.e34  | |#510=i$.s3.e82   |
            : d30 |#230.s3.e34  |  m30 |#530.s3.e34  | |#530=i$.s3.e82   |
            : d60 |#260.s3.e34  |  m60 |#560.s3.e34  | |#560=i$.s3.e82   |

 analysis 4 : d10 |#210.s4.e34  |  m10 |#510.s4.e34  | |#510=i$.s4.e82   |
            : d30 |#230.s4.e34  |  m30 |#530.s4.e34  | |#530=i$.s4.e82   |
            : d60 |#260.s4.e34  |  m60 |#560.s4.e34  | |#560=i$.s4.e82   |

                         inventory at the storage tanks:
                  plant 1 inventory             plant 2 inventory
           tank d10  tank d30  tank d60    tank m10  tank m30  tank m60
plan      :|#410.s1 ||#430.s1 ||#460.s1 |  |#610.s1 ||#630.s1 ||#660.s1 |
analysis 2:|#410.s2 ||#430.s2 ||#460.s2 |  |#610.s2 ||#630.s2 ||#660.s2 |
analysis 3:|#410.s3 ||#430.s3 ||#460.s3 |  |#610.s3 ||#630.s3 ||#660.s3 |
analysis 4:|#410.s4 ||#430.s4 ||#460.s4 |  |#610.s4 ||#630.s4 ||#660.s4 |
```

BEST-PLAN D I F I N T E R F A C E

BEST-PLAN supports linkages to other systems or programs. This may be accomplished in two directions:

DATA from BEST-PLAN to other programs or systems.

DATA from other programs or systems into BEST-PLAN, replacing the standard suggested procedure for data-entry, e.g. what-if scenario analysis may get the new resource prices and limits from a LOTUS 1-2-3 planning model.

This operation may be accomplished in two ways, first using REPORT to replace fields within an input text datafile(e.g. VISICALC's .VC files) and second using the well established DIF formatted datafiles, created automatically by BEST-PLAN during the SAVE SOLUTION operation.

.VC AND DIF INTERFACE using REPORT

The following examples shows how REPORT could be used to pass solution data to a VISICALC .VC file and/or a previously created DIF formatted file.

```
...................................................
 >C10:/FI@SUM(C5...C8)
 >B10:@SUM(B5...B8)
 >A10:"TOTAL:
 >C9:/-=
 >B9:/-=
 >C8:/FI(B8/B10)*100
 >B8:|#100=vl.e24   |          .......volume of #100
 >A8:/FR"|#100=nm       |   and name identification
 >C7:/FI(B7/B10)*100
 >B7:|#210=vl.e24   |          .......volume of #210
 >A7:/FR"|#210=nm           |   and name identification
 >C6:/FI(B6/B10)*100
...................................................
```

DIF formatted file:

```
..................................................
TABLE
0,1
"PRODUCTION"              ......title assigned
VECTORS
0,2                       ......no. of fields = 2
""
TUPLES
0,2                       ......no. of records = 2
""
DATA
0,0
""
-1,0
BOT
1,0                       ......name identification
"|#100=nm          |"           of #100
0,|#100=vl.e24|            ......volume of #100
V
-1,0
BOT
1,0                       ......name identification
"|#210=nm          |"           of #210
0,|#210=vl.e24|           ......volume of #210
V
-1,0
EOD
..................................................
```

Both files (.VC and/or DIF) may be created using the available SCREEN Editor or word processor. These files will be given the standard filenames for report templates. (i.e. Pnnnn.Frr) Report writing (Selection 1) is activated to create a Report using this template on any of the available SCREENS. The resulting file will be:

BP.SCn

n=(screen no.)-1 Corresponding RENAME may be necessary.

AUTOMATIC DIF INTERFACE

OUTPUT from BEST-PLAN: BEST-PLAN also supports an automatic DIF file creation during optimization. A DIF formatted datafile may be created by BEST-PLAN during the SAVE SOLUTION operation prior to storing the solution into its database. The DIF specification and data information may be established during the UTILITY procedure. Here the planner specifies the data contents and order of the DIF file to be automatically created.

The DIF file created by the SAVE SOLUTION operation will have the VISICALC column orientation (i.e. /S#S name C) and the following specifications:

```
DIF filename= PnnnnSss.DIF
               nnnn = PLAN,  ss = scenario

VECTORS =   the number of solution data  requested during the
            UTILITY procedure.

TUPLES  =   the number of resources, or variables with
            solution information.

LABELS  =   the datanames of the specified VECTORS,
            e.g.prices, lower limits, etc..

LABELS used are:
"RESOURCES"   = for the variables and resources
"DESCRIPTION" = for the name.identification
"VOLUME"      = for the solution volume
"PRICES"      = for the prices and costs
"INT.COST"    = for the propagated internal costs
"TOTAL.$$"    = for the totalcost (VOLUME*PRICE/or*INT.COST)
"INCENTIVE"   = for the reduced cost/shadow prices
"LOW.LMT"     = for the lower limit assigned to resource
"HIGH.LMT"    = for the upper limit assigned to resource

The "RESOURCES" data field will carry the variable numbers with
of the following format:

 #nnnn           #nnnn  =resource no. (variable no.)
```

The following will show how the file is created:

```
Header Section:   TABLE
                  VECTORS
                  TUPLES
                  LABELS

            indicate   the  datanames  here  for  the  corresponding
            VECTORS  in  the  data  section.  e.g.  PRICE,  LOW,  FIXED,
            etc..


Data Section  : DATA
                BOT

                .............V E C T O R S..............
                LABELS will contain datanames of VECTORS
```

"RESOURCES"	data/title	data/title
#nnnn	data	data
#nnnn	data	data
#nnnn	data	data
#nnnn	data	data
#nnnn	data	data

TUPLES

EOD

The first VECTOR specified should be the "RESOURCES" or resource/variable no. The "RESOURCES" vector will carry the variables with the following format:

```
 #nnnn           #nnnn  =resource no. (variable no.)
```

INPUT into BEST-PLAN: BEST-PLAN also supports data input for scenario analysis using pre-created DIF from another program or system. Resource/variables name identifications, prices and limits may be entered into BEST-PLAN from a DIF file. The DIF file will have the VISICALC column orientation (i.e. /S#S name C) and the following specification:

```
DIF filename= PnnnnDss.DIF
              nnnn = PLAN,  ss = scenario

VECTORS =   the number of data fields to be entered. This
            input facility can only support  the following
            data changes:

Name.identifications = assign name.id to resources and
                       variables(30-characters).
Prices and Costs     = assign new prices and costs to
                       resources and variables.
Limits: low, high and fixed
                     = assign new limits to resource and
                       variables.

TUPLES =    the number of resources, or variables with new
            information.

LABELS =    the datanames of the specified VECTORS, e.g.
            prices, lower limits, etc..

            LABELS that are valid are:
                  "RESOURCES","VARIABLES"
                 "DESCRIPTION"
                 "PRICE" "COST"
                 "L.L.", "LOW", "MINIMUM"
                 "U.L.", "UPPER", "MAXIMUM"
                 "FIXED", "SET", "HOLD"
```

The DIF file may specify datanames using the first TUPLE in the data section. The first TUPLE may contain string data equivalent to the labels as shown. This will allow the datanames to come from the spreadsheet as header information.

The following shows how the input DIF file should be created:

```
Header Section:   TABLE
                  VECTORS
                  TUPLES
                  LABELS

            indicated the datanames here for the corresponding
            VECTORS in the data section. e.g. PRICE, LOW, FIXED,
            etc..


Data Section  :DATA
                BOT

                 ............V E C T O R S.............
                LABELS will contain datanames of VECTORS
                      or the first TUPLE's VECTORS
```

"RESOURCES"	data/title	data/title
#nnnn	data	data
#nnnn	data	data
#nnnn	data	data
#nnnn	data	data
#nnnn	data	data

TUPLES

EOD

The first VECTOR specified should be the "RESOURCES" or resource/variable no. The "RESOURCES" vector will carry the variables with the following format:

```
 #nnnn           #nnnn  =resource no. (variable no.)
```

The option to read a DIF file for data-entry during scenario analysis is specified during the UTILITY operation. Here the planner merely specifies that a possible DIF is to be tested after the standard input (Pnnnn-ss.IN) has been read for input. The standard data filename for input during a scenario analysis is:

FILENAME: Pnnnn-ss.IN

nnnn = PLAN NO.
ss = SCENARIO NO.

If the DIF input file capability has been requested, BEST-PLAN will first read the above file (which can be empty) and then check the existence of the following file:

FILENAME: PnnnnDss.DIF

This file is then read as data-entry additions for the standard input data and reported on SCREEN 1. as the case may be.

Optimization will then proceed as normal and the SOLUTION DIF file is created on the SAVE SOLUTION operation. The solution DIF file will be given the following filename:

FILENAME: PnnnnSss.DIF

The previous UTILITY execution will specify the data contents and order of this file.

BEST-PLAN P L A N N I N G G U I D E

DATA-ENTRY

STEP-BY-STEP Material FLOW Representation Method

The following are the ENGLISH-LIKE data statements used to translate the FLOW DIAGRAM for input to BEST-PLAN. The data statements are all FREE-FORM, so the planner can incorporate spaces and comments to facilitate his personal style. There is no rule in respect to data ordering, so the planner can group data statements according to his liking. The language compiler REQUIRES that all DIRECTIVES (as explained in this manual) be CAPITALIZED. The LIST include the following:

```
INCLUDE, COPY
MACHINE, PROCESS, SECTION, PLANT, UNIT
RESOURCE, VARIABLES, MATERIALS, ACTIVITY
MAXIMIZE, MINIMIZE
(parameter data statements)
PRICES, VALUES
COSTS, EXPENSES
MINIMUM, LOW, L.L., LOWERB, LOW.LMT
MAXIMUM, HIGH, U.L., UPPERB, HIGH.LMT
FIXED, HOLD, SET
LIMITS, BOUNDS, RANGE
(equation or relational data statements)
USE, SPLIT, COMBINE, ACCUMULATE, INPUTS
POOL, INVENTORY, WIP
DISTRIBUTE, SUPPLY
APPLYCOST, ADDCOST
EQUATIONS, EQUATE, CONSTRAINTS, ROWS
XFERCOST, XFER$
TABLE, TABULATION
DATA
END, ENDATA
(removing or deleting relational data statements)
REMOVE, DROP, DELETE
(command preposition for relational data statements)
FROM, WITH, FOR, TO, MAKE, TOMAKE, INTO, OUTPUTS
(relational operators for equations)
<LE>, <LT>
<EQ>, EQUALS
<GE>, <GT>
<RN>, <RANGE>
```

IT IS THE INTENTION THAT THE INPUT DATA FOR THE PLAN ALSO SERVE AS THE DOCUMENTATION FOR FUTURE REFERENCES. THE PLANNER IS URGED TO USE COMMENTS FREELY AND TO ORDER HIS DATA TO BEST SUIT HIS PLANNING PREFERENCE.

STEP-BY-STEP material FLOW Representation

TERMINOLOGY

The economic or production plan will be presented in what BEST-PLAN refers to as a FLOW DIAGRAM. This flow diagram may represent any of the following:

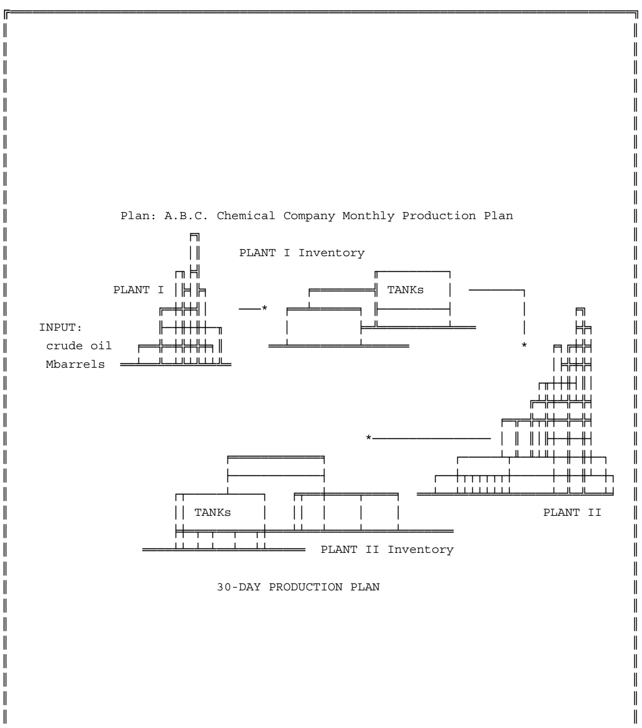
- processes, and/or machineries within a plant, mill, store
- an entire plant, mill, store
- a network of plants, mills, stores
- a network of plants, with distribution warehouses
- freight and transportations considerations to consumer stores or product outlets.

The diagram will illustrate all resources, expenses, raw materials and products involved in the plan. The level of detail will be dependent on the number of these resources. Nevertheless, some resources should be ignored if they have small, if not negligible, contribution to the overall estimated profit for us to make a good decision .

The demonstration problem will be represented in a schematic FLOW diagram. The production sequence will be divided into sections or processes. The section or process selections may be established by the following factors:

- Reporting requirements after the analysis
- Administrative considerations in the plants
- Recipe or operational differentials
- Physical orientation of the plants
- Other considerations

DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.



DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.

The planning period will be 30 days. During this period 900 million barrels of crude oil will be consumed first by plant #1 to produce semi-processed products and by-products. Operating cost are consumed and storage tanks will be considered. Plant #2 will further process the products to its refined state. Final inventory is maintained through limited capacity storage tanks. Find the most profitable combination of products the plants should produce, find the inventory status at month's end and the minimized overall production cost.

ANALYSIS SCENARIOS:

1. The most profitable production schedule for all three products.

Products Refined	Production Volumes
M10	20 to 35 million barrels
M30	20 to 35 million barrels
M60	10 to 30 million barrels

2. The production schedule if the product demands are fixed at specific amounts.

M10, M30	33 million barrels
M60	24 million barrels

3. With the same fixed production, consider the following inventory left over by last month's production.

	TANKS(plant 1)	TANKS(plant 2)
M10	40 mbbls	15 mbbls
M30	50 mbbls	15 mbbls
M60	50 mbbls	15 mbbls

4. Allow for next month's possible production stoppage. The final inventory for all products in TANKS(plant 2) must total between 45 to 50 million barrels.

DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.

PLANT #1 OPERATION

Plant 1 produces three types of refined oil (D10), D30 and D60) using three different operation processes.

TABLE I: Plant 1 Operation

All values per barrels of crude	Process 1	Process 2	Process 3
Operating Cost	$2.00	$2.00	$2.10
D10 Yield	0.1	0.1	0.0
D30 Yield	0.1	0.0	0.1
D60 Yield	0.0	0.1	0.1
Process (day)	0.03333	0.03333	0.03333
Yield Byproduct #1	0.04	0.04	0.04
Yield Byproduct #2	0.1	0.1	0.1

The refined oil (D10, D30, and D60) is stored into three tanks respectively (TankD10, TankD30 and TankD60).

The tanks have the following capacities and inventory limits.

	MIN	MAX
Tank D10	20 mbbl	80 mbbl
Tank D30	20 "	80 "
Tank D60	20 "	80 "

DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.

PLANT #2 OPERATION

Plant 2 further refines the oil (D10, D30 and D60) using again three different operating processes, producing (M10, M30 and M60) respectively.

TABLE II: Plant 2 Operation

All values per barrels of input	Process 1	Process 2	Process 3
Operating Cost	$0.60	$0.60	$0.60
M10 from D10	0.62	-	-
M30 from D30	-	-	-
M60 from D60	-	-	0.39
Process (day)	0.16667	0.16667	0.19231
Yield Byproduct #3	0.38	0.55	0.61
Capacity of Plant 2	6.0	6.0	5.2

These refined oil M10, M30 and M60 are kept in holding tanks prior to shipment.

The tanks have the following capacities and inventory limits.

	MIN	MAX
Tank M10	7 mbbl	25 mbbl
Tank M30	7 "	25 "
Tank M60	7 "	25

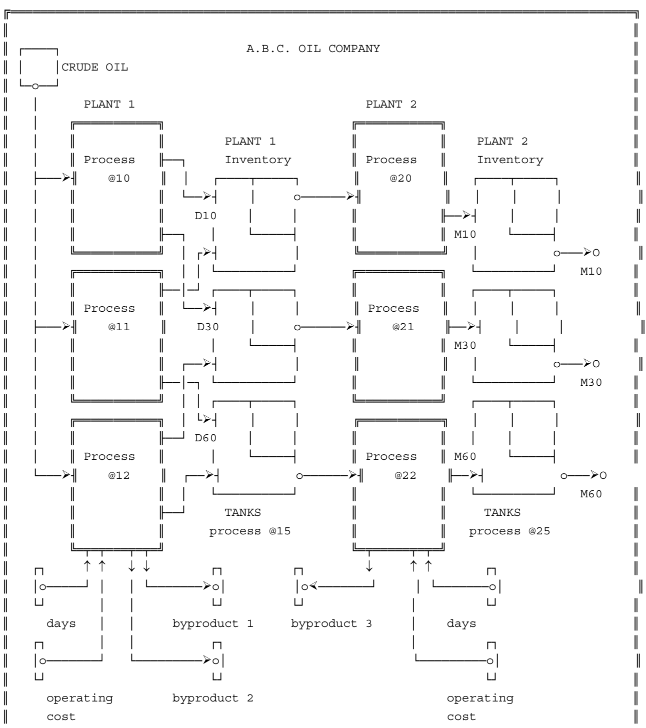
```
DEMONSTRATION PROBLEM:      A.B.C. Chemical Company
                            Produces three(3) oil based
                            products through 2 plants.
```

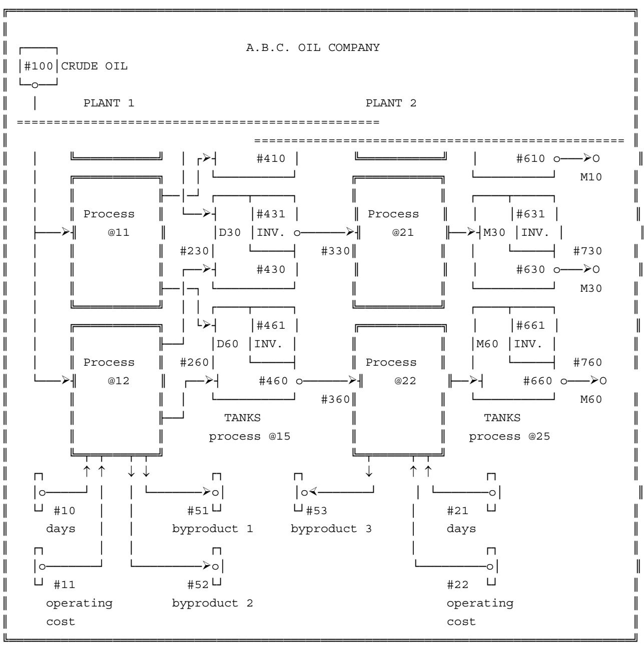
REPORTING REQUIREMENTS. In order to help us in our analysis, a report as shown is needed after an execution of the model.

```
    A.B.C. Chemical COMPANY                      todays date: 11/04/83

    cost summary report    PLAN:  1984 scenario:   1

    demonstration:         lubricating oil plant optimization

   scenario:     PLAN:  A.B.C.  Company  Production      master plan

   cost(minimized):  ($    ***,***,***)   infeasibility: **,***
     analysis date: **-**-**                     count: ***
              time: **:**:**

plant #1  cost summary       oil costs       op. costs       total cost

d10 & d30  *,***.**** mbl  $ ***,***,***. $ **,***,***.  $   ***,***,***.
d30 & d60  *,***.**** mbl  $ ***,***,***. $ **,***,***.  $   ***,***,***.
d10 & d60  *,***.**** mbl  $ ***,***,***. $ **,***,***.  $   ***,***,***.
  ----------------------------------------------------------------------
totals     *,***.**** mbl  $ ***,***,***. $ **,***,***.  $   ***,***,***.


plant #2  cost summary     plt #1 cost   plt #2 op. cost    total cost

m10 product *,***.**** mbl $ ***,***,***. $ **,***,***.  $   ***,***,***.
m30 product *,***.**** mbl $ ***,***,***. $ **,***,***.  $   ***,***,***.
m60 product *,***.**** mbl $ ***,***,***. $ **,***,***.  $   ***,***,***.
  ----------------------------------------------------------------------
totals      *,***.**** mbl $ ***,***,***. $ **,***,***.  $   ***,***,***.
  ----------------------------------------------------------------------
          total operating costs  (plant #1 & #2)         $   ***,***,***.

              products         volume          overall costs/mbbls

            m10 product        **,***.**          $ ***,***,***.
            m30 product        **,***.**          $ ***,***,***.
            m60 product        **,***.**          $ ***,***,***.
```

```
DEMONSTRATION PROBLEM:     A.B.C. Chemical Company
                           Produces three(3) oil based
                           products through 2 plants.
```

FLOW DIAGRAM





All resources (i.e. lines) consumed or produced within these processes are assigned a resource number (1-9999). The planner may assign this number arbitrarily or relational by following a certain convention.

i.e.	raw materials	500-599
	direct costs	700-799
	energy resources	200-299
	products	1000-1199

The number assigned to a specific resource will identify that resource throughout the model or plan. A name identification may be assigned to this resource for use in reporting and analysis later on. Up to 30 characters are allowed for name identifications.

Processes or sections are also assigned a number (1-999) and a corresponding name identification (30-characters).

All resources will be involved in a process, whether consumed or produced. A RESOURCE SHOULD NOT BE CONSUMED AND PRODUCED WITHIN THE SAME PROCESS. This will cause ambiguity within BEST-PLAN's mathematical modeling system.

Name identifications should be assigned once during data entry. Any reassignment will be treated as an update to the first one. Name identification may not contain slashes (/), commas (,), asterisks (*), equals (=), at-signs (@), semicolons (;), single-quotes ('), question mark(?), plus (+), minus(-), pound-signs (#), nor TABS.

The following shows all the processes and resources we have identified in the demonstration problem. It lists the assigned number and its corresponding 30-character name identifications.

DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.

Process No.	Name Identification
Plant 1 processes	
@10	d10 & d30 prod. plt 1
@11	d10 & d60 prod. plt 1
@12	d30 & d60 prod. plt 1
@15	holding tanks: after plt 1
Plant 2 processes	
@20	m10 production plt 2
@21	m30 production plt 2
@22	m60 production plt 2
@25	holding tanks: after plt 2

DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.

Resource No.	Name Identification
	Plant 1 processes
#10	plt 1 production days
#11	plt 1 operating cost
#100	crude oil (m bbls)
#210	d10 production plt 1
#230	d30 production plt 1
#260	d60 production plt 1
#51	byproduct 1
#52	byproduct 2
	Tanks after Plant 1
#411	tank d10 initial inv.
#431	tank d30 initial inv.
#461	tank d60 initial inv.
#410	tank d10 final inv.
#430	tank d30 final inv.
#460	tank d60 final inv.
	Plant 2 processes
#20	plt 2 production days
#21	plt 2 operating cost
#310	d10 to plt 2
#330	d30 to plt 2
#360	d60 to plt 2
#510	m10 production plt 2
#530	m30 production plt 2
#560	m60 production plt 2
#53	byproduct 3
	Tanks after Plant 2
#611	tank m10 initial inv.
#631	tank m30 initial inv.
#661	tank m60 initial inv.
#610	tank m10 final inv.
#630	tank m30 final inv.
#660	tank m60 final inv.
#600	total inventory tanks plt 2
#710	m10 total production
#730	m30 total production
#760	m60 total production

SUMMARY

BEST-PLAN'S MODELING CONCEPT AND TERMINOLOGY

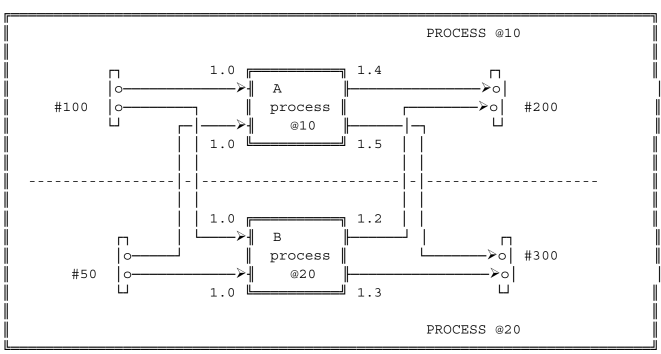
- A product FLOW DIAGRAM for a production cycle, distribution or logistic problem is used to facilitate input data creation for BEST-PLAN.

- The product FLOW DIAGRAM will be divided into unique sections or processes. A process number (1-999) and a 30 character name identification is assigned to these processes.

- Within a process, resources will be consumed or produced, but not both. This resource will also be assigned a number (1-9999) and a 30-character name identification.

- All resources and processes will be identified using this number in the plan and their corresponding name identifications used during the reporting and analysis.

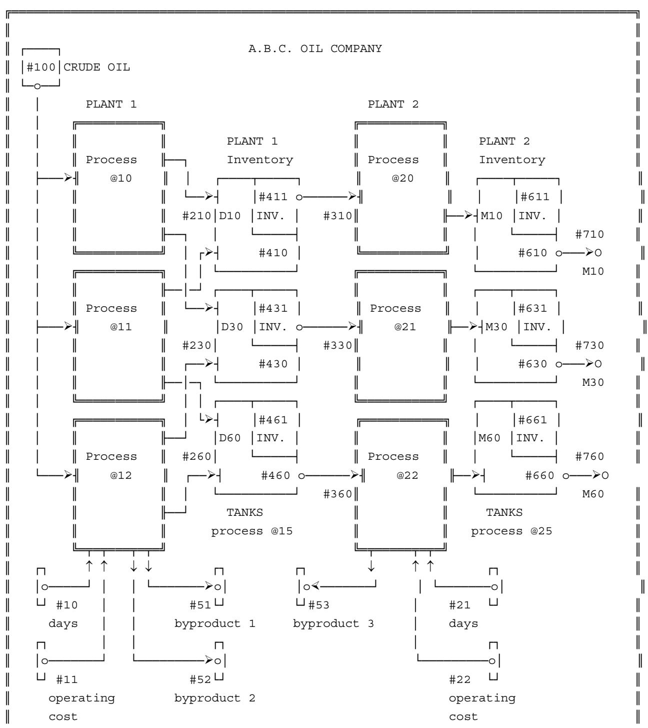
- The basic relational requirement of BEST-PLAN is to have a resource uniquely utilized within a process in relation to other resources, i.e.



A and B are two different processes since it utilizes #100 to make #200 and #300 with different yields.

DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.

Complete FLOW DIAGRAM



PROCESS IDENTIFICATION

Input data for BEST-PLAN is segmented by process or sections. This could be looked at as BLOCKS of information. It starts with a PROCESS declaration and terminates with another PROCESS statement. An END statement may also be used to terminate a PROCESS statement.

The PROCESS declaration is as follows:

PROCESS @ppp = name identification (30-characters)

@ppp = process number (1-999)

PROCESS,MACHINE,SECTION,UNIT are synonymous and can be used interchangeably.(see. UTILITY)

A typical block of input data for BEST-PLAN is as follows:

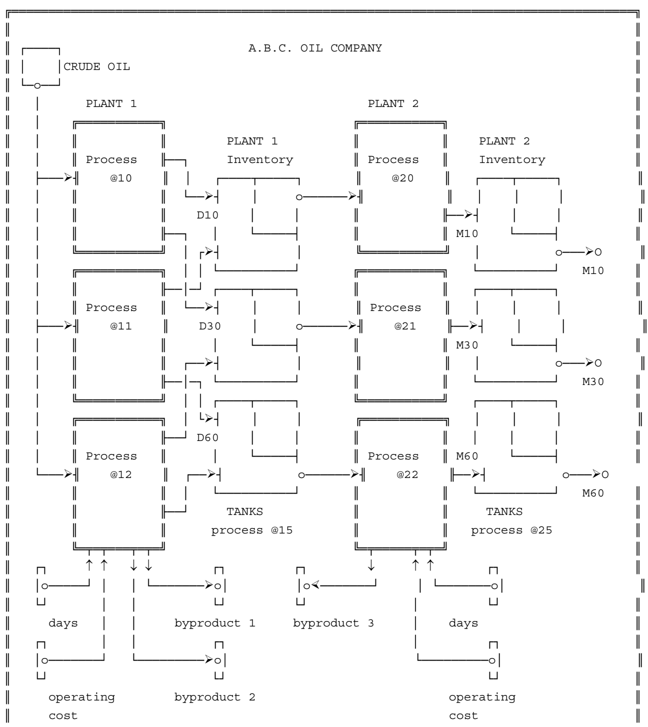
```
'.......................................................
   '
   '     process to produce D10 and D30 in plant 1
   '
 PROCESS @10 = (d10 & d30) production plnt.1

   ......   other resources and relational data   .........
......                                       .........

   '
   '
 PROCESS @11 = (d30 & d60) production plnt.2

   ......   other resources and relational data   .........
......                                       .........
 END
'.......................................................
```

The at-sign(@) must precede the process numbers to differentiate it from resource numbers.

The single-quote(') is data entry facility for comments and information. It allows the data-entry procedure to ignore the rest of the line. Remarks and comments are not inputs to the system, but are self-help information for the planner.

DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.

The figure illustrates the problem divided into SECTIONS or processes. The general rule is that A RESOURCE MAY NOT BE CONSUMED AND PRODUCED WITHIN THE SAME PROCESS.



RESOURCE IDENTIFICATION

A resource will have some information associated with it (i.e. price, limits/demands, volume). Some of these are entered by the planner during data-entry and some computed by BEST-PLAN after the optimization of the plan. Some resources are defined as intermediate resources. These resources are produced by a process and consumed by another within the plan. Their importance to reporting may be considered minimal since they are not sold or bought. On the other hand, if BEST-PLAN is used to handle production scheduling, these resources may have to be identified. The minimal informational requirement for a resource is the resource number (1-9999) and name identification.

Resource data declaration is as follows:

    RESOURCE #nnnn = name identification(30-characters)

        #nnnn = resource number (1-9999)

RESOURCE,MATERIALS are synonymous and can be used interchangeably.(see. UTILITY)

Additional input information for a resource are as follows:

- Price or cost. Cost is a negative price. Data is important for resources bought or sold outside our plan or FLOW DIAGRAM. All resources carry a unit of measure. The prices and costs should be prices and costs per corresponding unit of measure.

- Limits(low/upper). Planning limits control how much the resource will vary for the planning period. All resources default to a no-limit(free) assignment, which means that any positive value is acceptable.

Solution data are computed by BEST-PLAN after the plan goes thru the optimization. A resource will have the following solution data:

- Volume. This amount shows the resource contribution to the most profitable operation. This volume is controlled by any possible low and/or upper limit imposed by the planner.

- Internal Cost. Some resources are products of processes, which accumulate costs during their production. The internal cost is the total cost accumulated per unit of this resource. This value is computed by the cost propagation program of BEST-PLAN.

- Total price. This is the overall value of the resource, which is the volume times the unit price or internal cost depending on whether it is a product or a resource bought from outside the plan.

- Reduced Cost. Also known as shadow price or INCENTIVE. If the resource has reached its upper limit, as imposed by the planner, the incentive price shows how much the profit could be increased or decreased by increasing or decreasing the upper limit for this resource by one (1.0) unit.

ACTIVITY IDENTIFICATION

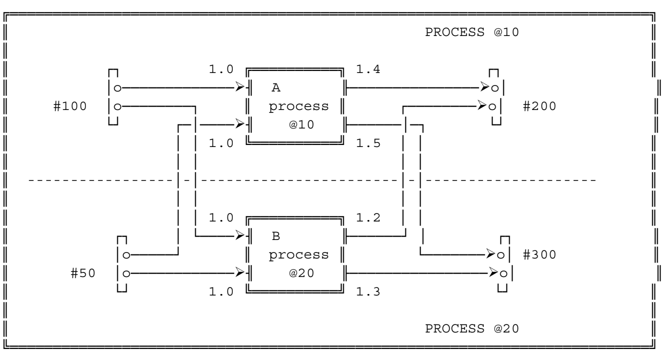
ACTIVITY is the performance of a given resource within a process, whether it is consumed or produced. Activities will have the same information discussed previously. i.e. price/cost, internal cost, total price, volume, limits (low/upper) and reduce cost. An activity is different from a resource in that it only pertains to a specific process or section of the plan. Therefore, a resource is the aggregated and/or the averaged of all its activities.

The ACTIVITY is identified as follows:

```
      @ppp #nnnn  = Activity number

            @ppp = process number  (1-999)
           #nnnn = resource number (1-9999)
```

Activity data declaration is as follows:

```
      ACTIVITY @ppp #nnnn = name identification(30-characters)
```




```
  i.e. (#100) equals (@10 #100) + (@20 #100)
  PROCESS @10, activities are (@10 #100),(@10 #50)
                              (@10 #200),(@10 #300)
  PROCESS @20, activities are (@20 #100),(@20 #50)
                              (@20 #200),(@20 #300)
```

Relationship between RESOURCES in a PROCESS (ACTIVITY)

The relationship between processes, resources and activities are established by the FIXED FLOW relational data statements to be discussed later. The following will show a typical EXAMPLE:

```
'.......................................................
'      plant  1  processes
'.......................................................

  PROCESS @10=d10 & d30 prod. plt 1

      USE #100/crude oil (m bbls)       =1.0
           #10/plt  1 production days   =0.033333
           #11/plt  1 operating cost    =2000.
     MAKE #210/d10 production plt 1     =0.1
          #230/d30 production plt 1     =0.1
           #51/byproduct  1             =0.04
           #52/byproduct  2             =0.1;
```

```
          the process no. = @10
          the resources   = (#100),(#10),(#11),
                          = (#210),(#230),(#51),(#52)
          the activities  = (@10 #100),(@10 #10),(@10 #11)
                          = (@10 #210),(@10 #230)
                          = (@10 #51), (@10 #52)

     the relationship:   base activity = (@10 #100)

     1.0 * (@10 #100) = 1.0      * (@10 #100)
     1.0 * (@10 #10)  = 0.033333 * (@10 #100)
     1.0 * (@10 #11)  = 2000.0   * (@10 #100)
     1.0 * (@10 #210) = 0.1      * (@10 #100)
     1.0 * (@10 #230) = 0.1      * (@10 #100)
     1.0 * (@10 #51)  = 0.04     * (@10 #100)
     1.0 * (@10 #52)  = 0.1      * (@10 #100)
```

Only the base activity (@10 #100) will be included in the MATRIX(table of equations) to minimize the number of variables. The rest of the activities will be re-computed by the solution expansion program after the optimization using the same relationship.

OTHER CONSIDERATION

The system uses the following naming convention for the data mentioned:

```
                                  "RAW" Solution LIST
                                       and Report Template

      code . . . . . . .            #nnnn
         @ppp  =process no.         @ppp #nnnn
         #nnnn =resource no.
      name identification           NM
      (assigned by planner)
      price or cost(unit)           P$
      internal cost                 I$
      low limit                     LL
      upper limit                   UL
      fixed limit(low=upper)        FX
      volume                        VL
      total price                   T$
      reduce cost(shadow price)     R$
```

```
DEMONSTRATION PROBLEM:       A.B.C. Chemical Company
                             Produces three(3) oil based
                             products through 2 plants.
```

The following will show an excerpt from the "RAW" Solution Listing:

```
    ..........master plan no:  1984   scenario no:  1 ...........

           max:profit   FUNCTIONAL      -199.6112E+05
                        OBJECTIVE       *************

          ITERATIONS:  79              OPTIMIZATION DATE: 05-01-1988
     INFEASIBILITIES:   0                           TIME: 12:00:00

 V A R I A B L E S (COLUMNS)
 ....................................................................
      ACTIVITY :  VL= volume   R$= reduced cost   I$= internal cost
                  P$= price    LL= lower limit    UL= upper limit
 .....NAME....AT................ACTIVITY.............................
 #10         BS  VL= 29.9997   UL= 30
 #100        LL  VL= 900  R$= 24281.07  I$=-25000  P$=-25000  FX= 900
 #11         BS  VL= 1800000   I$=-1   P$=-1
 #20         UL  VL= 30   R$= 66664.23   UL= 30
 #21         BS  VL= 103380.6   I$=-1   P$=-1
 #210        BS  VL= 39.95702   I$= 269999.9
 #230        BS  VL= 90   I$= 270000
 #260        BS  VL= 50.04299   I$= 270000
 #310        BS  VL= 32.25806   I$= 269999.9
 #330        BS  VL= 90   I$= 270000
 #360        BS  VL= 50.04299   I$= 270000
 #410        BS  VL= 7.698949   I$= 269999.9   UL= 80
 #51         BS  VL= 36   I$= 674999.9   P$= 12000
 #510        BS  VL= 20   I$=-433548.2
 #52         BS  VL= 90   I$= 270000   P$= 22000
 #53         BS  VL= 92.28428   I$=-501867.9   P$= 22000
 #530        BS  VL= 40.5   I$=-597333.3
 #560        BS  VL= 19.51676   I$=-689230.9
 #630        BS  VL= 5.499997   I$=-593333.4   UL= 25
 #710        LL  VL= 20   R$= 5404.403   I$=-430645   LL= 20   UL= 35
 #730        UL  VL= 35   I$=-593333.3   LL= 20   UL= 35
 #760        BS  VL= 19.51676   I$=-684615.5   LL= 10   UL= 30
 @10 #100    BS  VL= 399.5701   I$= 27000
 @12 #100    BS  VL= 500.4298   I$= 27000
 @15 #210    BS  VL= 39.95702   I$=-269999.9
 @15 #230    BS  VL= 90   I$=-270000
 @15 #260    BS  VL= 50.04299   I$=-270000
 ....................................................................
```

PRICES AND LIMITS DATA STATEMENTS

The planner can enter prices and limits for resources in the model as well as the base activities created by the FIXED FLOW relational data statements. The planner can also group them together in one data statement.

PRICE/COST

The PRICE data statement is intended for resources bought or sold outside the scope of the PLAN.

```
PRICES #nnnn = value, #nnnn = value
            .........#nnnn = value;

       nnnn = resource numbers (1-9999)
      value = unit price or cost
              (positive) sold outside
              (negative) bought outside
```

(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator

PRICES,VALUES are synonymous and can be used interchangeably.(see. UTILITY)

COSTS,EXPENSES are synonymous and can be used to replace PRICE if the values are negative. This will reverse the sign (positive to negative) internally.

```
'.........................................................
'                         prices
'.........................................................

PRICES  #100=-25000.     '...crude oil
        #11=-1.0         '...Plant 1 operating cost
        #21=-1.0         '...Plant 2 operating cost

        #51=12000.       '...byproduct 1 SOLD
        #52=22000.       '...byproduct 2 SOLD
        #53=22000.       '...byproduct 3 SOLD
   @10 #100=100.;        '...$100 credit for process @10

'.........................................................
```

PRICE/COST

Resource prices and name identifications can be combined together using the PRICES and COSTS data statements. Their corresponding order is immaterial.

```
PRICES #nnnn /name identification=value
       #nnnn =name identification/value
       #nnnn =value/name identification
       #nnnn /value=name identification;

        nnnn = resource numbers (1-9999)
        name identification (30-characters)
       value = unit price or cost

(/) slashes and (=) equals are used as separators
(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator
```

```
'.........................................................
'                        prices
'.........................................................

  PRICES  #100=-25000./crude oil (m bbls)
          #11/-1.0=Plant 1 operating cost
          #21/Plant 2 operating cost=-1.0

          #51=12000./byproduct 1
          #52=22000./byproduct 2
          #53=22000./byproduct 3
     @10 #100=100./credit for process @10;

'.........................................................
```

LIMITS

The planner specifies product demand, material availability, time availability for processes and inventory status through the LIMITS data statements.

```
LIMITS #nnnn = value1:value2,#nnnn = value1:value2
             ................#nnnn = value1:value2;

       nnnn = resource numbers (1-9999)
     value1 = low limit
     value2 = upper limit
```

(:)colons are used to separate low& upper limits
(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator

LIMITS,RANGE,BOUNDS are synonymous and can be used interchangeably.(see. UTILITY)

```
'.......................................................
'         demands on production
'.......................................................
'
  LIMITS  #710=20. : 35.  '..refined product M10
          #730=20. : 35.  '..refined product M30
          #760=10. : 30.  '..refined product M60
      @12 #100=20. : 60.; '..limit process @12 operation

'.......................................................
```

LIMITS

Resource limits and name identifications can be combined together using the LIMITS data statements. Their corresponding order is immaterial.

```
      LIMITS #nnnn /name identification=value1:value2
             #nnnn =name identification/value:value2
             #nnnn =value1:value2/name identification
             #nnnn /value1:value2=name identification;

              nnnn = resource numbers (1-9999)
              name identification (30-characters)
            value1 = low limit
            value2 = upper limit

     (/) slashes and (=) equals are used as separators
     (,) commas and (next-line) are used as separators
     (;) semicolon is used as terminator
```

```
   '.........................................................
   '          demands on production
   '.........................................................
   '
     LIMITS  #710=20. : 35./refined product M10
             #730/refined product M30=20. : 35.
             #760=10. : 30./refined product M60
         @12 #100=20. : 60./process @12 operation;

   '.........................................................
```

FIXED

Both lower and upper limits may be the same value for certain resources. The planner may want a very tight control on production. The FIXED data statement may be used.

```
FIXED #nnnn = value, #nnnn = value
           .........#nnnn = value;

      nnnn = resource numbers (1-9999)
     value = both low and upper limit
```

(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator

FIXED,HOLD,SET are synonymous and can be used interchangeably.(see. UTILITY)

```
'.......................................................
'       initial tank levels from last month's production
'       Plant 1
'.......................................................

    FIXED  #411=14.0      '..tank D10
           #431= 7.0      '..tank D30
           #461= 8.5;     '..tank D60

'.......................................................
```

FIXED

Resource FIXED limits and name identifications can be combined together using the FIXED data statements. Their corresponding order is immaterial.

```
      FIXED #nnnn /name identification=value
           #nnnn =name identification/value
           #nnnn =value/name identification
           #nnnn /value=name identification;

            nnnn = resource numbers (1-9999)
            name identification (30-characters)
           value = low and upper limit

     (/) slashes and (=) equals are used as separators
     (,) commas and (next-line) are used as separators
     (;) semicolon is used as terminator
```

```
    '.......................................................
    '       initial tank levels from last month's production
    '       Plant 1
    '.......................................................

       FIXED  #411=14.0/tank D10 initial contents
              #431= 7.0/tank D30 initial contents
              #461= 8.5/tank D60 initial contents;

    '.......................................................
```

L.L.
U.L.

The lower and upper limits can be entered separately using the L.L. or the U.L. data statements respectively.

```
L.L. #nnnn = value, #nnnn = value
          .........#nnnn = value;

U.L. #nnnn = value, #nnnn = value
          .........#nnnn = value;

     nnnn = resource numbers (1-9999)
    value = low or upper limit

(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator
```

L.L.,LOW,MIN,MINIMUM are synonymous and can be used interchangeably.(see. UTILITY)

U.L.,UPPER,MAX,MAXIMUM are synonymous and can be used interchangeably.(see. UTILITY)

```
'........................................................
'       set the tank low limits
'       inventory requirements
'........................................................

  L.L.  #610 = 25.        '..tank M10
        #630 = 25.        '..tank M30
        #660 = 25.;       '..tank M60

'........................................................
```

L.L.
U.L.

Resource L.L. or U.L. limits and name identifications can be combined together using the L.L. or U.L. data statements. Their corresponding order is immaterial.

```
L.L. #nnnn /name identification=value
     #nnnn =name identification/value
     #nnnn =value/name identification
     #nnnn /value=name identification;

      nnnn = resource numbers (1-9999)
      name identification (30-characters)
     value = low or upper limit
```

(/) slashes and (=) equals are used as separators
(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator

```
'.........................................................
'       set the tank low limits
'       inventory requirements
'.........................................................

   L.L.  #610/tank M10 final inv.=25.
         #630/tank M30 final inv.=25.
         #660/tank M60 final inv.=25.;

'.........................................................
```

A typical RESOURCE,PRICES and LIMITS data statement are as follows:

```
'.......................................................
'   production:   from plant #2
'   refined products demand per month
'.......................................................


RESOURCE #710=m10 total production
RESOURCE #730=m30 total production
RESOURCE #760=m60 total production

'.......assign product prices...........................
'............................. from marketing BOB

PRICES #710 = 16.0,   #730 = 15.0,   #760 = 14.0;

'.......assign limits monthly demand....................

LIMITS   #710 = 20.0 : 35.0
         #730 = 20.0 : 35.0
         #760 = 10.0 : 30.0;

'........................................................
```

The above example shows that spaces can be used sparingly throughout the data statements; that a carriage return (newline) works similar to the commas; that the RESOURCE data line does not need a semicolon terminator, and that the colon sign is used to separate the lower and the upper limit in the LIMITS command.

In all data statements, spaces can be used for aesthetic considerations. A new line also works like a comma and can be used as the separator. Some special characters are interchangeable so the planner can develop his own style of entering data into BEST-PLAN.

In summary, data statements follow the general rules:

- Single Quotes (') are used for comments. The rest of the line is ignored when detected.

- Semicolon (;) terminates a data statement. This termination is important for PRICES, LIMITS and the relational data statements to be discussed later. On an ERROR, the remaining part of the data statement is ignored by scanning the following lines until the next semicolon.

- Commas (,), TABs or a new line (carriage return) are used as separators between data-to-resource assignments.

- Spaces can be used sparingly for aesthetic consideration. These are ignored, except when used inside a name identification assignment where it is accepted as part of the 30-character maximum capability.

- Minus (-) is used to indicate negative values in the PRICES data statement. A corresponding COSTS data line will have positive values that are internally converted to negative values.

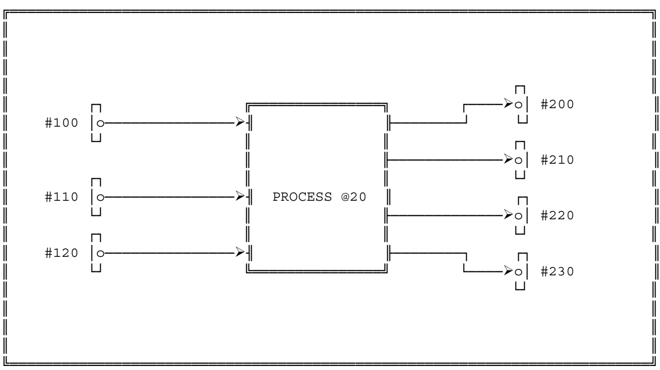
- Equals (=) and slashes (/) are used as data element separators for a resource within data-to-resource assignment.

- Colon (:) is also used to separate the lower limit and the upper limit in the LIMITS data statement.

- Name identifications should not contain the following special characters: ('), (,), (TAB), (/), (=), (#), (@), (?), (+), (-), (*)

RELATIONAL DATA STATEMENTS

Within a process, a resource is consumed, combined with other resources, distributed to other processes, and possibly accumulating costs and value-added. This resource-to-resource relationship is identified through the relational data statements. This relationship is best exemplified by the following FLOW DIAGRAM.



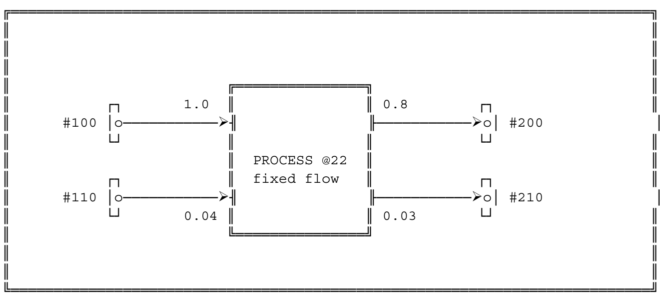
The above diagram shows that three resources #100, #110 and #120 are combined together and distributed as #200, #210, #220 and #230. This FLOW diagram is represented into BEST-PLAN by the following relational data statements:

```
'..................................................
 PROCESS @20 = distribution of steam
'..................................................

     USE  #100, #110, #120
 TO MAKE  #200,  #210, #220,  #230;
'..................................................
```

Here USE is the relational verb and TOMAKE is the relational preposition. Note that the relational data statement is within a PROCESS block.

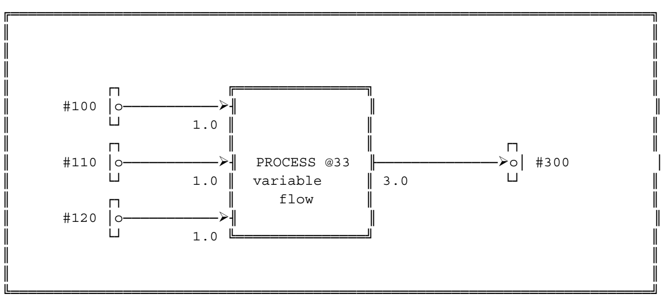
A relational data statement could either be unconditional or fixed flow (i.e. a recipe) or conditional or variable flow (i.e. routing by demand). The following example shows the differences:

Unconditional (FIXED FLOW)



Here 1.0 units of #100, and 0.04 units of #110 will produce 0.8 units of #200 and 0.03 units of #210.

Conditional (VARIABLE FLOW)



Here resource #300 can come from three sources #100, #110 and #120. There is no fixed relationship between the resources, therefore the system will select whichever of the sources (i.e. #100, #110, #120) or any combination which will be most profitable. Same as: (#100) + (#110) + (#120) equals (#300)

USE/MAKE

Relational data declarations are as follows:

Unconditional (FIXED FLOW)

```
      USE   #nnnn = value, #nnnn = value,...
           #nnnn /value,  #nnnn /value,...
      MAKE  #nnnn = value, #nnnn = value,...
           #nnnn /value,  #nnnn /value,....;

            nnnn = resource numbers (1-9999)
           value = recipe value, yields
                   (amounts needed/produced)

             USE  is the relational verb
             MAKE is the relational preposition
```

There must be at least one space between the last value and MAKE (the relational preposition) if a comma is not used.

Conditional (VARIABLE FLOW)

```
      USE   #nnnn, #nnnn, #nnnn,.....
     MAKE  #nnnn, #nnnn, #nnnn,.....;

            nnnn = resource numbers (1-9999)
            no values are entered

     (/) slashes and (=) equals are used as separators
     (,) commas and (next-line) are used as separators
     (;) semicolon is used as terminator
```

USE,SPLIT,COMBINE are synonymous and can be used interchangeably. The planner may use whichever suits the relational verb he selected. (see. UTILITY)

FROM,WITH,TO,FOR,MAKE,TOMAKE,INTO,OUTPUT are synonymous and can be used interchangeably. The planner may use whats appropriate with the verb above.(see. UTILITY)

USE/MAKE

USE/MAKE relational data declarations and name identifications can be combined together using the USE/MAKE relational data statements. Their corresponding order is immaterial.

```
      USE   #nnnn /name identification=value
           #nnnn =name identification/value
           #nnnn =value/name identification
           #nnnn /value=name identification
      MAKE  #nnnn /name identification=value
           #nnnn =name identification/value
           #nnnn =value/name identification
           #nnnn /value=name identification;

            nnnn = resource numbers (1-9999)
            name identification (30-characters)
           value = recipe value, yields
                   (amounts needed/produced)

     (/) slashes and (=) equals are used as separators
     (,) commas and (next-line) are used as separators
     (;) semicolon is used as terminator
```

DEMONSTRATION PROBLEM: A.B.C. Chemical Company Produces three(3) oil based products through 2 plants.

```
    '.........................................................
    '          lubricant production from crude oil
    '
    '               plant  1  processes
    '.........................................................
          PROCESS @10=d10 & d30 prod. plt 1

              USE #100/crude oil (m bbls)       =1.0
                   #10/plt  1 production days   =0.033333
                   #11/plt  1 operating cost    =2000.
             MAKE #210/d10 production plt 1     =0.1
                  #230/d30 production plt 1     =0.1
                  #51/byproduct  1              =0.04
                  #52/byproduct  2              =0.1;

    '.........................................................
```

DEMONSTRATION PROBLEM: A.B.C. Chemical Company Produces three(3) oil based products through 2 plants.

```
'.........................................................
       PROCESS @11=d10 & d60 prod. plt 1

           USE #100=1.0,   #10=0.033333,  #11=2000.
          MAKE #210=0.1
               #260/d60 production plt 1     =0.1
                #51=0.04,  #52=0.1;

'.........................................................

  PROCESS @12=d30 & d60 prod. plt 1

          USE #100=1.0,   #10=0.033333,  #11=2000.
         MAKE #230=0.1,  #260=0.1
               #51=0.04,  #52=0.1;

'      operational constraint  total days = 30

         MAXIMUM  #10=30.;

'.........................................................

'      plant  2  processes

       PROCESS @20=m10 production plt 2

          USE #310=1.0
               #20/plt 2 production days     =0.166667
               #21=plt 2 operating cost      =600.
          MAKE #510/m10 production plt 2     =0.62
               #53/byproduct  3              =0.38;

'.........................................................

       PROCESS @21=m30 production plt 2

          USE #330=1.0
               #20=0.166667
               #21=600.
          MAKE #530/m30 production plt 2     =0.45
               #53=0.55;

'.........................................................
```

DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.

```
'.........................................................

        PROCESS @22=m60 production plt 2

      USE #360=1.0
           #20=0.192307
           #21=600.
     MAKE #560/m60 production plt 2      =0.39
           #53=0.61;


'       operational constraint  total days = 30

          MAXIMUM  #20=30.;
'.........................................................
'      holding tanks:  after plant  2

         PROCESS  @25=holding tanks: after plt 2

        POOL #610/tank m10 final inv.
            FROM #510,#611/tank m10 initial inv.
              TO #710/m10 total production;
'.........................................................
```

USE/MAKE

Additional relational data statements are available for special applications. They are derivatives of the USE/MAKE statements. They also follow the general syntax or convention used for the main relational data statement. These will be discussed later in this section. They are as follows:

```
-   POOL   #nnnn
    FROM   #nnnn, #nnnn, #nnnn,..
      TO   #nnnn, #nnnn, #nnnn...;

-   APPLY  #nnnn
      TO   #nnnn = value, #nnnn = value,.;

-   PROCESS    @ppp
    DISTRIBUTE #nnnn
         TO    #nnnn = value, #nnnn = value,.;
```

SUMMARY: Difference between FIXED FLOW and VARIABLE FLOW

VARIABLE FLOW:

1. There is no fix and specific relationship between the resources involved in the process. The performance (volume) of the resources will depend on the overall profitability of the model.

2. The relationship between the resources is true for the entire MODEL and not specific to a section or process.

3. Except for the "DISTRIBUTE" version of the variable flow, no activities are created by this data statement.

4. EXAMPLE: COMBINE #100, #200, #300 INTO #100; means, (#100) + (#200) + (#300) equals (#1000) this applies to the entire MODEL.

FIXED FLOW:

1. There is a fix and specific relationship between the resources involved in the process. This relationship is true only for this process and not for the entire MODEL. i.e. another process may have an entirely different fix relationship between the same resources.

2. Activities (@ppp #nnnn) are created for this data statement. The fix relationship between the resources will be computed relative to a base activity. The base activity involves a resource selected using the following priority:
   a. resource with a value= 1.0 (INPUT or OUTPUT). The side of lesser number of resources is selected.
   b. if none, the first resource (INPUT or OUTPUT). The side of lesser number of resources is selected.
   c. if equal, the first resource in the INPUT side.

3. EXAMPLE: PROCESS @10

```
               COMBINE #100=0.55, #200=1.00, #300=.86
                  INTO #1000=1.00;
base activity = @10 #1000
relationships: 0.55 * (@10 #100) = 1.0 * (@10 #1000)
               1.00 * (@10 #200) = 1.0 * (@10 #1000)
               0.86 * (@10 #300) = 1.0 * (@10 #1000)
```

POOL or INVENTORY

This data statement is a special version of the variable flow used to represent a pool, inventory, warehouse, holding area, and/or a work in process. The inventory level at the pool can then be limited by the same LIMITS statement, and the pool treated as a regular resource. (i.e. it will carry a resource number, name identification, cost of operation, etc.)

```
POOL #nnnn
       FROM #nnnn, #nnnn, #nnnn,...
         TO #nnnn, #nnnn, #nnnn,...;

             nnnn = resource numbers (1-9999)
```

(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator

POOL,INVENTORY are synonymous and can be used interchangeably.(see. UTILITY)

The first #nnnn is the pool resource FROM is used to indicate the sources of the pool(i.e. the supply) and TO is used to indicate where its used or distributed.

```
'.......................................................
'      holding tanks:  after plant  1

        PROCESS  @15=holding tanks: after plt 1


       POOL #410  FROM #210, #411
                    TO #310;

       POOL #430  FROM #230, #431
                    TO #330;

       POOL #460  FROM #260, #461
                    TO #360;

'.......................................................
```

POOL/INV

POOL relational data declarations and name identifications can be combined together using the POOL relational data statements. Their corresponding order is immaterial.

```
 POOL  #nnnn /name identification
FROM  #nnnn /name identification
      #nnnn =name identification
TO    #nnnn /name identification
      #nnnn =name identification;

       nnnn = resource numbers (1-9999)
       name identification (30-characters)

(/) slashes and (=) equals are used as separators
(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator
```

```
'........................................................

'      holding tanks:  after plant  1

        PROCESS  @15=holding tanks: after plt 1

       POOL #410/tank d10 final inv.
           FROM #210,#411/tank d10 initial inv.
              TO #310/d10 to plt  2;

       POOL #430/tank d30 final inv.
           FROM #230,#431/tank d30 initial inv.
             TO #330/d30 to plt  2;

       POOL #460/tank d60 final inv.
           FROM #260,#461/tank d60 initial inv.
             TO #360/d60 to plt  2;

'........................................................
```

DISTRIBUTE

This data statement is a special version of the variable flow used to represent a distribution process. This statement must be preceded by a PROCESS declaration. ACTIVITIES will be created to represent the distribution of the resource. A typical use of this data statement is a planning MODEL where a product is to be distributed to its markets.

```
 PROCESS     @ppp
DISTRIBUTE  #nnnn
        TO #nnnn, #nnnn, #nnnn,...;

            ppp  = process number   (1-999)
            nnnn = resource numbers (1-9999)
```

(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator

DISTRIBUTE,SUPPLY are synonymous and can be used interchangeably.(see. UTILITY)

Corresponding activities (@ppp #nnnn) will be created to represent the distribution.

```
'.......................................................
'      plant  1 production of PROCESS A

       PROCESS  @44         'Plant.1 Chicago
     DISTRIBUTE  #400       'Product A
            TO  #1          'LosAngeles product A
                #2          'Detriot product A
                #3          'Seattle product A
                #4          'Dayton product A
                #5          'Minneapolis product A;
'.......................................................
```

Activities created:

@44 #1 = Product A to LosAngeles
@44 #2 = Product A to Detriot
@44 #3 = Product A to Seattle
@44 #4 = Product A to Dayton
@44 #5 = Product A to Minneapolis
@44 #400 = Plant.1 Product A (base activity)

DISTRIBUTE

DISTRIBUTE relational data declarations and name identifications can be combined together using the DISTRIBUTE relational data statements. Their corresponding order is immaterial.

```
      PROCESS @ppp/name identification
     DISTRIBUTE  #nnnn /name identification
             TO  #nnnn /name identification
                 #nnnn =name identification
                 #nnnn /name identification
                 #nnnn =name identification;

                   ppp = process number   (1-999)
                  nnnn = resource numbers (1-9999)
                  name identification (30-characters)

     (/) slashes and (=) equals are used as separators
     (,) commas and (next-line) are used as separators
     (;) semicolon is used as terminator
```

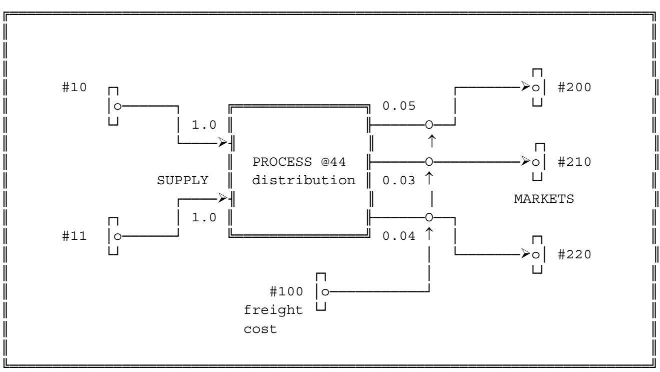
```
   '........................................................

   '      plant  1 production of PRODUCT A

          PROCESS  @44=Plant.1 Chicago

         DISTRIBUTE  #400/Product A
                 TO  #1/LosAngeles product A
                     #2/Detriot product A
                     #3/Seattle product A
                     #4/Dayton product A
                     #5/Minneapolis product A;

   '........................................................
```

APPLY

This data statement is special version of the fixed flow used to designate added costs or added values for a given resource. A good application of this statement is the application of freight charges to resources moving from one location to another. Handling charges may also be represented by this data statement.

```
      APPLY #nnnn
        TO #nnnn = value, #nnnn = value,...
           #nnnn /value,  #nnnn /value,...;

            nnnn = resource numbers (1-9999)
           value = value to be applied per unit

     (/) slashes and (=) equals are used as separators
     (,) commas and (next-line) are used as separators
     (;) semicolon is used as terminator
```

APPLYCOST,ADDCOST are synonymous and can be used interchangeably.(see. UTILITY)



APPLY #100 TO #200 /0.05, #210 /0.03, #220 /0.04;

Here the / sign is used instead of the equal (=) sign for a better representation.

APPLY

APPLY relational data declarations and name identifications can be combined together using APPLY relational data statements. Their corresponding order is immaterial.

```
      APPLY #nnnn /name identification=value
        TO #nnnn /name identification=value
           #nnnn =name identification/value
           #nnnn =value/name identification
           #nnnn /value=name identification;

            nnnn = resource numbers (1-9999)
            name identification (30-characters)
           value = value to be applied per unit

     (/) slashes and (=) equals are used as separators
     (,) commas and (next-line) are used as separators
     (;) semicolon is used as terminator
```

```
'.........................................................
'      apply freight to product A
'.........................................................

   APPLY #100/Freight charges(AIR)
      TO @44 #1=0.05/LosAngeles product A
         @44 #2=0.03/Detriot product A
         @44 #3=0.04/Seattle product A

      COST #100=1.00;'.....$-1.00 unit cost
'.........................................................
```

APPLY data statements are NOT INCLUDED in the mathematics involved during the OPTIMIZATION. They are computed after the optimization, using the solution produced. The convenience offered by the APPLY data statement can be seen when increasing/or decreasing overall freight cost by a percentage. This is accomplished by changing the unit cost of the freight resource.

XFERCOST

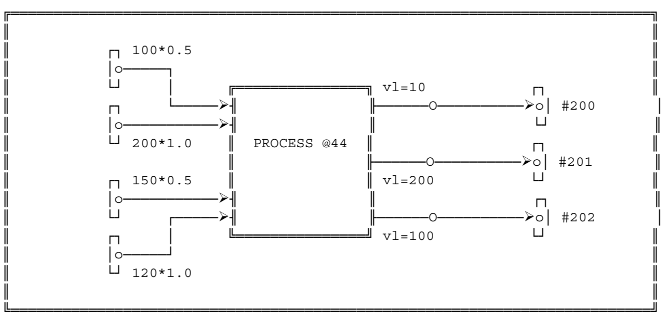
INTERNAL COST PROPAGATION

After the optimization, BEST-PLAN propagates all costs through the network (economic plan). This will allow us to track down the accumulated cost of resources throughout the system. (The total cost of the process is equally allocated to the products.) The cost is propagated as follows:

For VARIABLE FLOW: (i.e. pool,distribute) The cost of the input resources are added to determine the average cost per unit. This averaged Internal Cost (per unit) is then passed to the output resources.

For FIXED FLOW: The cost of the input resources are added according to their contributions and the accumulated cost is distributed equally to the output resources.

Cost propagation may be DISABLED when its not applicable using the UTILITY program.(see. UTILITY)



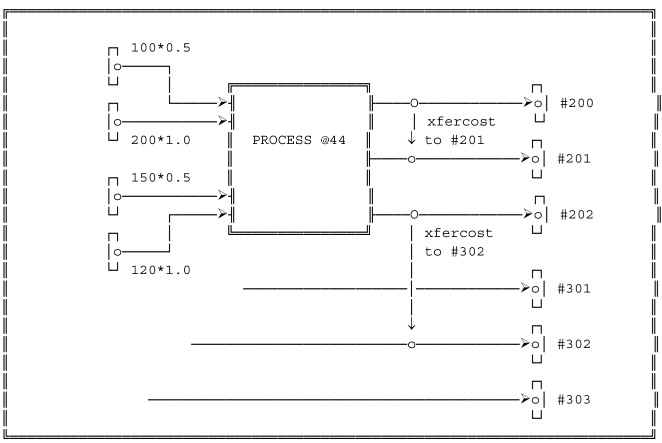

```
      Totalcost=100*(-0.5)+200*(-1.0)+150*(-0.5)+120*(-1.0)
               = -445.00
Cost(#200)= -445.00/3 = -148.33; I$(200) = -$14.833
Cost(#201)= -445.00/3 = -148.33; I$(201) = -$ 0.742
Cost(#202)= -445.00/3 = -148.33; I$(202) = -$ 1.483
```

The individual costs of #200, #201, and #202 are then again propagated through the next down stream processes.

XFERCOST

CONTROLLING INTERNAL COST PROPAGATIONS

The cost of resources could be transferred or added to another resource. A special data entry statement has been created to facilitate this requirement. The transfer involves the overall internal cost per unit of a given resource. Remember: positive PRICES or REVENUES are not propagated.



The illustration shows:

- The internal cost of #200 (I$) is to be transferred (or added to) the internal cost of #201.
- The internal cost of #202 is to be transferred (or added to) the internal cost of #302.

The following statements should be included in the input data to BEST-PLAN:

```
              XFERCOST #200 to #201;
                       #202 to #302;
```

The internal cost of #200 is then transferred to #201, and #200 will carry no internal cost (I$=0) as it moves through the network. #201 will carry the accumulated internal cost.

XFERCOST

XFERCOST data statements are NOT INCLUDED in the mathematics involved during the OPTIMIZATION. They are computed after the optimization, using the solution produced.

```
XFERCOST #nnnn
     TO #nnnn = value, #nnnn = value,...
        #nnnn /value,  #nnnn /value,...;

         nnnn = resource numbers (1-9999)
        value = value to be applied per unit
```

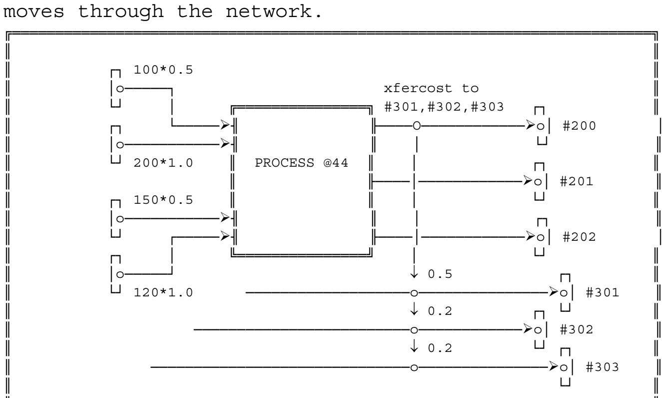
(/) slashes and (=) equals are used as separators
(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator

XFERCOST,XFER$ are synonymous and can be used interchangeably.(see. UTILITY)

The following data entry statement will apportion the cost to more than one resource:

```
XFERCOST #200 to #301/0.5, #302/0.2, #303/0.2;
```

The cost #200 is transferred as follows: 50% to #301, 20% to #302 and 20% to #303. And the remaining 10% will be carried by #200 as it moves through the network.



XFERCOST

XFERCOST relational data declarations and name identifications can be combined together using XFERCOST relational data statements. Their corresponding order is immaterial.

```
      XFERCOST #nnnn /name identification=value
           TO #nnnn /name identification=value
              #nnnn =name identification/value
              #nnnn =value/name identification
              #nnnn /value=name identification;

               nnnn = resource numbers (1-9999)
               name identification (30-characters)
              value = value to transfer

     (/) slashes and (=) equals are used as separators
     (,) commas and (next-line) are used as separators
     (;) semicolon is used as terminator
```

```
'.........................................................
'
'      Transfer Cost of product A to product B,B9 & B10
'.........................................................

    XFERCOST #200/Product A
          TO #301=0.5/Product B
             #302=0.2/Product B9
             #303=0.2/Product B10;

'.........................................................
```

ADDITIONAL DATA STATEMENTS

Three (3) additional data statements are supplied to support the planner during his scenario analysis as well as for general convenience during data entry, namely:

1. INCLUDE data statement. This allows the planner to include or copy data from another scenario or a common data update file. Nesting is not supported. A file which has been INCLUDEd cannot INCLUDE another FILE. The data format is as follows:

```
INCLUDE  'P1984-03.IN
INCLUDE  'B:COMMON.DTA
```

2. REMOVE data statement. This allows the planner to remove a relational data statement from the master PLAN for the purpose of analyzing a scenario or "WHAT-IF" analysis.

3. TABLE data statement. This allows the planner to develop the MODEL or PLAN conveniently using tables instead of redundantly stated data statements. The TABLE data statement applies to PRICES and LIMITS as well as relational statements like, USE, POOL, APPLY, DISTRIBUTE, EQUATIONS and XFERCOST plus their corresponding synonyms.

TABLE data statements are set up with three (3) sections:

```
TABLE
...the relational data statement to use....

DATA
...........................................
...the corresponding data arranged ........
...into a table or tabulations.............
...........................................
END
```

REMOVE/DELETE

Removing or deleting a relational data statement from the master PLAN during a scenario analysis is accomplished by the following STEPS:

1. Copy the original relational data statements from the master PLAN into the scenario update FILE.

2. Corresponding PROCESS declaration should also be copied if it applies. This is specially true for FIXED FLOW relational data statements.

3. Insert the word REMOVE or DELETE in front of the relational data statement.

SAMPLE:

```
'........................................................
'      plant  1  processes
'........................................................
        PROCESS @10=d10 & d30 prod. plt 1

 REMOVE  USE #100/crude oil (m bbls)      =1.0
             #10/plt  1 production days   =0.0333333
             #11/plt  1 operating cost    =2000.
        MAKE #210/d10 production plt 1    =0.1
             #230/d30 production plt 1    =0.1
             #51/byproduct  1             =0.04
             #52/byproduct  2             =0.1;

        PROCESS @11=d10 & d60 prod. plt 1

 REMOVE  USE #100=1.0,   #10=0.0333333,  #11=2000.
        MAKE #210=0.1
             #260/d60 production plt 1     =0.1
             #51=0.04,  #52=0.1;
 END
  '........................................................
```

REMOVE,DELETE,DROP are synonymous and can be used interchangeably.(see. UTILITY)

The END statement may be used to terminate a process before the next set of data.

TABLE

The TABLE data statement is a very powerful mechanism to minimize the redundancy of similar data statements for some MODELS. This is accomplished by following the STEPS:

1. Start with the word TABLE and then followed by the PRICE, LIMITS or relational data statement to be used for the TABLE. This may be any of the previously discussed data statements like:

PRICES/COSTS	USE/MAKE
FIXED	POOL/INVENTORY
L.L./U.L.	DISTRIBUTE
LIMITS	APPLY
EQUATE/EQUATIONS	XFERCOST

2. The question mark(?) will represent the positions within this relational data statement which will be replaced by data in the following table. The statement is accordingly terminated with the semicolon (;). These replaceable positions may be a resource, a data value or two(2) data values (LIMITS).
   i.e. USE #?, #100=? MAKE #?=?, #50=?;

   This line requires five(5) data values, in the following order:
   1. a resource number
   2. a data value
   3.,4. a resource number and data value
   5. a data value

3. This is then followed by the DATA section, with numbers listed in tabular form. The numbers within the DATA table MUST be separated by commas(,), TABS or carriage return. For convenience and aesthetic, comments lines may be added freely. DATA section for the above sample may be:

```
DATA
    #100,     5.00,     #10,  2.0,   55.6
    #101,     5.55,     #11,  2.3,   80.0
END
```

4. Terminate the data section with the END statement.

EXAMPLES USING TABLE

```
' The DISTRIBUTIONS: source logs allocated
' ---------------------------------------
     SECTION @10/LOG.SALES Allocations

' ..................................................................
  TABLE   USE #?=    ?,      #51=?,  FOR  #63=?,   #73=?;
  DATA
' ..................................................................
'          Resource  Units     Barker    Log.Sales  Log.Sales
'           No.                Hrs.      Disposal   Profit
' ..................................................................

         #8,    1.0,      0.050,      1.0,      180.
         #9,    1.0,      0.050,      1.0,      180.
        #15,    1.0,      0.063,      1.0,      180.
        #16,    1.0,      0.059,      1.0,      180.
        #17,    1.0,      0.056,      1.0,      180.
        #18,    1.0,      0.053,      1.0,      180.
        #19,    1.0,      0.050,      1.0,      180.
' ..................................................................
  END

     SECTION @20/PLYWOOD Allocations

' ..................................................................
  TABLE
    USE #?= ?,      #51=?, FOR #62=?,  #72=?,    #64=?,    #74=?;
  DATA
' ..................................................................
'  Resource  Units     Barker   Plywood    Plywood   Lumber     Lumber
'    No.                Hrs.    Disposal   Profit    Disposal   Profit
' ..................................................................

      #8,    1.0,       0.053,    0.9,       156.,     0.1,        5.9
      #9,    1.0,       0.050,    0.9,       160.,     0.1,        5.9
     #15,    1.0,       0.063,    0.9,       134.,     0.1,        5.9
     #16,    1.0,       0.059,    0.9,       138.,     0.1,        5.9
     #17,    1.0,       0.056,    0.9,       141.,     0.1,        5.9
     #18,    1.0,       0.053,    0.9,       147.,     0.1,        5.9
     #19,    1.0,       0.050,    0.9,       152.,     0.1,        5.9
     #23,    1.0,       0.071,    0.9,       112.,     0.1,        5.9
```

TABLE

TABLE is very convenient for distribution MODEL using the DISTRIBUTE data statements.

Consider the following example: PRODUCT is used instead of RESOURCE, PLANT is used instead of PROCESS.

```
'.........................................................
'        the product is Distributed to the markets
'.........................................................

   PRODUCT #100=Product AA

PLANT @1=Flint
DISTRIBUTE #100  TO   #1/Chicago,  #2/Cleveland,  #3/Dayton,
                      #4/Minneapolis;

PLANT @2=Jamesville
DISTRIBUTE #100  TO   #1,          #2,            #3,
                      #4;

PLANT @3=St.Louis
DISTRIBUTE #100   TO  #1,          #2,            #3,
                      #4;

'.........................................................
'                shipping  cost
'.........................................................

'      to    CHICAGO    CLEVELAND    DAYTON      MINNEAPOLIS

'FLINT...........Plant @1
    COST @1 #1=27.00, @1 #2=23.00, @1 #3=31.00, @1 #4=69.00;
'JAMESVILLE......Plant @2
    COST @2 #1=10.00, @2 #2=45.00, @2 #3=40.00, @2 #4=32.00;
'ST.LOUIS........Plant @3
    COST @3 #1=30.00, @3 #2=54.00, @3 #3=35.00, @3 #4=57.00;
'.........................................................
```

The MODEL using TABLE is as follows:

```
'.........................................................
'         the product is Distributed to the markets
'.........................................................

   PRODUCT #100=Product AA

TABLE
   DISTRIBUTE  #?  to  #?, #?, #?, #?;
DATA
'
'                 Product                MARKETS
PLANT @1=Flint
                      #100,      #1,     #2,     #3,    #4
PLANT @2=Jamesville
                      #100,      #1,     #2,     #3,    #4
PLANT @3=St.Louis
                      #100,      #1,     #2,     #3,    #4
END

'.........................................................
'                shipping  cost
'.........................................................

TABLE
   COST    #1=?,     #2=?,     #3=?,     #4=?;
DATA
   '      to    CHICAGO     CLEVELAND   DAYTON    MINNEAPOLIS
   'FLINT
                 @1/27.00,   @1/23.00,  @1/31.00,  @1/69.00
   'JAMESVILLE
                 @2/10.00,   @2/45.00,  @2/40.00,  @2/32.00
   'ST.LOUIS
                 @3/30.00,   @3/54.00,  @3/35.00,  @3/57.00;

END
'.........................................................
```

SUMMARY: DATA-ENTRY

- Data associated to a resource as entered by the planner is as follows:
  - resource number (1-9999)
  - name identification (30 characters)
  - planning limits
  - prices or costs

- Data associated to a resource as computed by BEST-PLAN after optimization are as follows:
  - volume, production contribution
  - reduce cost(shadow prices)
  - total price
  - internal cost

- These data are represented in the raw solution listed as follows:
  - price = P$
  - volume = VL
  - lower limit = LL
  - upper limit = UL
  - fixed limit = FX
  - reduce cost = R$

- Data-entry statements can be FREE-FORM, and may involve multiple lines. Comments can be added utilizing the Single-Quotes (') as the indicator. Spaces can be used sparingly throughout the data line.

- Name identification may contain spaces, but may not contain the following:

(/) slashes	(,) commas
(=) equal sign	(;) semicolon
(@) at-sign	(*) asterisk
(') single-quote	(?) question mark
(-) minus sign	(+) plus sign
(#) pound-sign	TABs

- A new line (carriage-return) is treated as a comma. All price, limit and relational data statements must terminate with the semicolon (;).

- Relational data statements could either be unconditional (FIXED FLOW) or conditional (VARIABLE FLOW). Resources may be consumed or produced in a process by using any of the relational data statements.

- Additional relational data statements for special applications are as follows:

  POOL. For establishing inventory levels, warehouses and the like. The inventory level is then treated as a resource.

  APPLY. For the accumulation or accruing of cost or value during the production cycle. i.e. handling charges.

  DISTRIBUTE. For distribution problems to propagate a product to its markets. It automatically creates the necessary activities.

- Name identification can be combined with the price, limit and relational data statements. The special characters equals (=) and slashes (/) can be used interchangeably to separate the name identification from the values as needed.

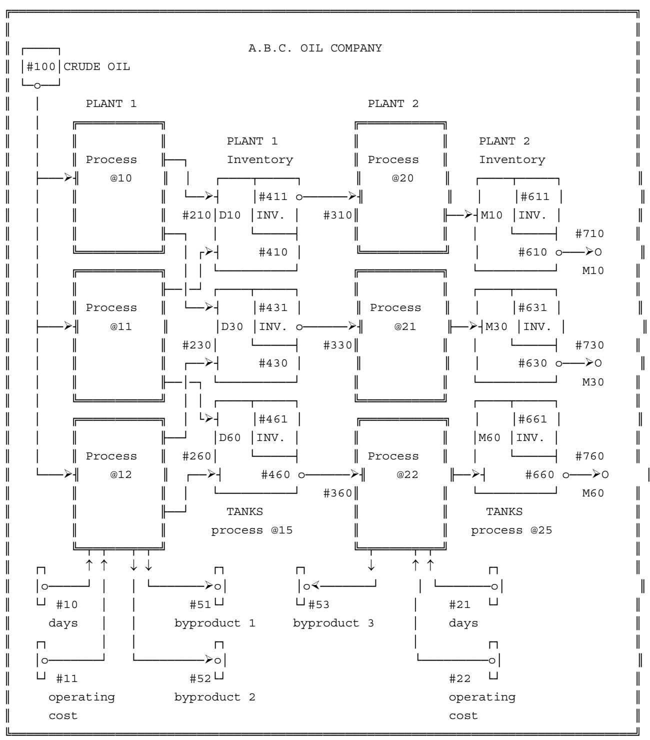
- Data statements are entered in BLOCKS preceded by the PROCESS date statement and terminated by another PROCESS statement or an END statement.

- Data statements can be ordered anyway the planner wants them. The basic idea is for the planner to develop his own planning language using the flexibility of the data entry media.

- The cost propagated through the network is called INTERNAL COST. This cost accounting may be transferred to other resources using the XFERCOST data statements. This will allow proper determination of product margins based on certain accounting practices.

DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.

Complete FLOW DIAGRAM



DEMONSTRATION PROBLEM: A.B.C. Chemical Company
Produces three(3) oil based
products through 2 plants.

PLAN: Complete INPUT DATA List P1984-01.IN

The following data was developed using the STEP-by-STEP FLOW Representation Method. Note that the MODEL was divided into processes.

```
'-------------------------------------------------------
'
'        A.B.C. Chemical Company
'
'-------------------------------------------------------
'      plant  1  processes
  PROCESS @10=d10 & d30 prod. plt 1

      USE #100/crude oil (m bbls)       =1.0
           #10/plt  1 production days   =0.033333
           #11/plt  1 operating cost    =2000.
     MAKE #210/d10 production plt 1     =0.1
          #230/d30 production plt 1     =0.1
           #51/byproduct  1             =0.04
           #52/byproduct  2             =0.1;

  PROCESS @11=d10 & d60 prod. plt 1

      USE #100=1.0,   #10=0.033333,  #11=2000.
     MAKE #210=0.1
          #260/d60 production plt 1     =0.1
           #51=0.04,  #52=0.1;

  PROCESS @12=d30 & d60 prod. plt 1

      USE #100=1.0,   #10=0.033333,  #11=2000.
     MAKE #230=0.1,  #260=0.1
           #51=0.04,  #52=0.1;
'.......................................................
  operational constraints  total days = 30
'.......................................................
    MAXIMUM  #10=30.;
```

```
'      holding tanks:  after plant  1
  PROCESS  @15=holding tanks: after plt 1

  POOL #410/tank d10 final inv.
      FROM #210,#411/tank d10 initial inv.
        TO #310/d10 to plt  2;

  POOL #430/tank d30 final inv.
      FROM #230,#431/tank d30 initial inv.
        TO #330/d30 to plt  2;

  POOL #460/tank d60 final inv.
      FROM #260,#461/tank d60 initial inv.
        TO #360/d60 to plt  2;
'..........................................................
'   set the tank limits and specify tank contents (inventory)
     MAXIMUM   #410 = 80.,   #430 = 80.,   #460 = 80.;
     FIXED     #411 = 0.0,   #431 = 0.0,   #461 = 0.0;
'..........................................................
'      plant  2  processes
  PROCESS @20=m10 production plt 2

     USE #310=1.0
          #20/plt 2 production days     =0.166667
          #21=plt 2 operating cost      =600.
    MAKE #510/m10 production plt 2      =0.62
          #53/byproduct  3              =0.38;

  PROCESS @21=m30 production plt 2

     USE #330=1.0
          #20=0.166667
          #21=600.
    MAKE #530/m30 production plt 2      =0.45
          #53=0.55;

  PROCESS @22=m60 production plt 2

     USE #360=1.0
          #20=0.192307
          #21=600.
    MAKE #560/m60 production plt 2      =0.39
          #53=0.61;
'..........................................................
'      operational constraints  total days = 30
    MAXIMUM  #20=30.;
```

```
 '      holding tanks:  after plant  2
   PROCESS  @25=holding tanks: after plt 2

   POOL #610/tank m10 final inv.
       FROM #510,#611/tank m10 initial inv.
         TO #710/m10 total production;

   POOL #630/tank m30 final inv.
       FROM #530,#631/tank m30 initial inv.
         TO #730/m30 total production;

   POOL #660/tank m60 final inv.
       FROM #560,#661/tank m60 initial inv.
         TO #760/m60 total production;
 '..........................................................
 '   set the tank limits and specify tank contents (inventory)
     MAXIMUM   #610 = 25.,   #630 = 25.,   #660 = 25.;
     FIXED     #611 = 0.0,   #631 = 0.0,   #661= 0.0;
 '..........................................................
 '          demands on production

    LIMITS  #710=20. : 35.  '..refined product M10
            #730=20. : 35.  '..refined product M30
            #760=10. : 30.; '..refined product M60
 '.........................................................
 '          limit the amount of crude oil
     FIXED  #100=900.;
 '.........................................................
 '                        prices

   PRICES  #100=-25000.     '...crude oil
            #11=-1.0        '...Plant 1 operating cost
            #21=-1.0        '...Plant 2 operating cost

            #51=12000.      '...byproduct 1 SOLD
            #52=22000.      '...byproduct 2 SOLD
            #53=22000.;     '...byproduct 3 SOLD
 '.........................................................
```

PRODUCTION SCHEDULING PROBLEM:

A.B.C. Chemical Company
Processes three(3) oil based
products through 2 plants.

SCREEN 2: Parsed input data display

This display will contain any possible input data errors detected during phase 1.

```
\'\..................................................\
\'\       changes to master plan:  analysis 4\
\'\..................................................\
\
\'\       fix demands for m10 at 33 mbbls\
\'\                       m30 at 33 mbbls\
\'\                       m60 at 24 mbbls\
\'\\
FIXED\\#\710\=\33.\,\\#\730\=\33.\,\\#\760\=\24.\;\\
\
\'\..................................................\
\'\        set initial tank levels\
\'\        plant  1 also specify unit price\
\
FIXED\\#\411\=\40.\,\\#\431\=\50.\,\\#\461\=\50.\;\\
\
\'\..................................................\
\'\        plant  2\
\'\\
\\FIXED\\#\611\=\15.\,\\#\631\=\15.\,\\#\661\=\12.\;\\
\
\'\..................................................\
\'\  limit total outstanding inv. at tanks m10,m30,m60 \
\
PROCESS\\@\25\\
\
EQUATE\\#\600\/\total inventory tanks plt 2\
<EQ>\\#\610\+\\#\630\+\\#\660\;\\
\
LIMIT\\#\600\=\45.:50.\;\\
\
```

PRODUCTION SCHEDULING PROBLEM:

A.B.C. Chemical Company
Processes three(3) oil based
products through 2 plants.

SCREEN 3: "RAW" Solution LIST

This display will contain the Linear programming solution in typical LP orientation.

```
    ..........master plan no:  1984   scenario no:  4 ............

              *** R A W   S O L U T I O N (LIST) ***


    ......S O L U T I O N ( INFEAS. )......

         max: profit   INFEASIBLE       311.5366E-02
                       ..TOTAL...      *************


           THE FOLLOWING ARE AFFECTED BY THE INFEASIBILITIES

......NAME.........ACTIVITY......LOWER LIMIT....UPPER LIMIT..........

 #560               8.855767       0             ..none..
 #660               0              0              25
 #661               12             12             12
 #760               24             24             24

ROW:   e@25 #660      <EQ>  RHS= -3.144233
   + 1*(#560) - 1*(#660) + 1*(#661) - 1*(#760)

......................................................................


    ......S O L U T I O N ( INFEAS. )......

         max: profit   FUNCTIONAL      -200.8759E+05
                       OBJECTIVE       *************

         ITERATIONS:   36              OPTIMIZATION DATE: 05-01-1988
    INFEASIBILITIES:    1                          TIME: 12:00:00
```

```
V A R I A B L E S (COLUMNS)

......................................................................
     ACTIVITY :  VL= volume   R$= reduced cost   I$= internal cost
                 P$= price    LL= lower limit    UL= upper limit
......NAME....AT................ACTIVITY..............................

#10         BS VL= 29.99997  UL= 30
#100        UL VL= 900  I$=-25000  P$=-25000  FX= 900
#11         BS VL= 1800000  I$=-1  P$=-1
#20         UL VL= 30  R$= 2.028242  UL= 30
#21         BS VL= 105903.8  I$=-1  P$=-1
#210        BS VL= 90  I$=-67500
#230        BS VL= 37.29291  I$=-67500
#260        BS VL= 52.70709  I$=-67500.01
#310        BS VL= 69.35484  I$=-46730.77
#330        BS VL= 84.44444  I$=-28837.07
#360        BS VL= 22.7071  I$=-34639.56
#410        BS VL= 60.64516  I$=-46730.77  UL= 80
#411        LL VL= 40  FX= 40
#430        BS VL= 2.848465  I$=-28837.07  UL= 80
#431        LL VL= 50  FX= 50
#460        UL VL= 80  I$=-34639.56  UL= 80
#461        LL VL= 50  FX= 50
#51         BS VL= 36  I$=-168750  P$= 12000
#510        BS VL= 43  I$=-38653.85
#52         BS VL= 90  I$=-67500  P$= 22000
#53         BS VL= 86.6506  I$=-38513.91  P$= 22000
#530        BS VL= 38  I$=-33374.52
#560        ** VL= 8.855767  I$=-45948.16
#610        UL VL= 25  R$= .2058906  I$=-29015.9  UL= 25
#611        UL VL= 15  R$= .5452108  FX= 15
#630        BS VL= 20  I$=-24406.89  UL= 25
#631        UL VL= 15  R$= .7511014  FX= 15
#661        ** VL= 12  R$= 1  FX= 12
#710        LL VL= 33  R$= .5452108  I$=-29015.9  FX= 33
#730        LL VL= 33  R$= .7511014  I$=-24406.89  FX= 33
#760        ** VL= 24  R$= 1  I$=-19837.12  FX= 24
@10 #100    BS VL= 372.9291  I$=-27000  UL= 900
@11 #100    BS VL= 527.0709  I$=-27000  UL= 900
@20 #310    BS VL= 69.35484  I$=-47330.77
@21 #330    BS VL= 84.44444  I$=-29437.07
@22 #360    BS VL= 22.7071  I$=-35239.56
#600        LL VL= 45  R$= .7511014  I$=-27379.18  LL= 45  UL= 50
```

```
O T H E R (ACTIVITIES)

......................................................................
     ACTIVITY :
                 VL= volume     P$= price        I$= internal cost
......NAME....AT................ACTIVITY..............................

@10 #10     EX VL= 12.43096
@11 #10     EX VL= 17.56901
@10 #11     EX VL= 745858.2  I$=-1  P$=-1
@11 #11     EX VL= 1054142  I$=-1  P$=-1
@20 #20     EX VL= 11.55916
@21 #20     EX VL= 14.0741
@22 #20     EX VL= 4.366734
@20 #21     EX VL= 41612.91  I$=-1  P$=-1
@21 #21     EX VL= 50666.66  I$=-1  P$=-1
@22 #21     EX VL= 13624.26  I$=-1  P$=-1
@10 #210    EX VL= 37.29291  I$=-67500
@11 #210    EX VL= 52.70709  I$=-67500
@10 #230    EX VL= 37.29291  I$=-67500
@11 #260    EX VL= 52.70709  I$=-67500.01
@10 #51     EX VL= 14.91716  I$=-168750  P$= 12000
@11 #51     EX VL= 21.08284  I$=-168750  P$= 12000
@20 #510    EX VL= 43  I$=-38653.85
@10 #52     EX VL= 37.29291  I$=-67500  P$= 22000
@11 #52     EX VL= 52.70709  I$=-67500  P$= 22000
@20 #53     EX VL= 26.35484  I$=-38513.91  P$= 22000
@21 #53     EX VL= 46.44444  I$=-38513.91  P$= 22000
@22 #53     EX VL= 13.85133  I$=-38513.91  P$= 22000
@21 #530    EX VL= 38  I$=-33374.52
@22 #560    EX VL= 8.855768  I$=-45948.16
```

```
C O N S T R A I N T S (ROWS)

.......................................................................
     ACTIVITY :
                             D$= dual price    RH= right-hand-side
......NAME.....................ACTIVITY..............................

 max: profit D$= 0  RH= 2.008759E+07
demand#10    D$=-1  RH= 20
demand#100   D$=-1.000026  RH= 527.0709
demand#11    D$= 1  RH= 1800000
demand#20    D$=-3.028242  RH= 69.35484
demand#21    D$= 1  RH= 105903.8
demand#310   D$= 1  RH= 69.35484
demand#330   D$= 1  RH= 84.44444
demand#360   D$= 1.000001  RH= 22.7071
e@15 #410    D$=-1  RH= 29.99997
e@15 #430    D$=-.9999999  RH= 60.64516
e@15 #460    D$=-.9999988  RH= 527.0709
e@25 #610    D$= .4547893  RH= 22.7071
e@25 #630    D$= .2488987  RH= 84.44444
e@25 #660    D$= 0  RH= 3.144233
supply#210   D$= 1  RH= 90
supply#230   D$= .9999999  RH= 37.29291
supply#260   D$= .9999988  RH= 52.70709
supply#51    D$= 1  RH= 36
supply#510   D$= 1.545211  RH= 43
supply#52    D$= 1  RH= 90
supply#53    D$= 1  RH= 86.6506
supply#530   D$= 1.751101  RH= 38
supply#560   D$= 2  RH= 8.855767
e@25 #600    D$=-.2488987  RH= 77.15154
.......................................................................
```

PRODUCTION SCHEDULING PROBLEM:

A.B.C. Chemical Company
Processes three(3) oil based
products through 2 plants.

SCREEN 4: EQUATIONS as formulated by BEST-PLAN

This display will contain the Linear programming formulations.

```
..........master plan no:  1984   scenario no:  4 ............

....updated master...  M A T R I X   (T A B L E A U)

...V A R I A B L E S...(COLUMNS) =  39

| #10        | #100       | #11        | #20        | #21        |
| #210       | #230       | #260       | #310       | #330       |
| #360       | #410       | #411       | #430       | #431       |
| #460       | #461       | #51        | #510       | #52        |
| #53        | #530       | #560       | #610       | #611       |
| #630       | #631       | #660       | #661       | #710       |
| #730       | #760       | @10 #100   | @11 #100   | @12 #100   |
| @20 #310   | @21 #330   | @22 #360   | #600       |

...C O N S T R A I N T S..(ROWS) =  25

|  max: profit| demand#10  | demand#100 | demand#11  | demand#20  |
| demand#21  | demand#310 | demand#330 | demand#360 | e@15 #410  |
| e@15 #430  | e@15 #460  | e@25 #610  | e@25 #630  | e@25 #660  |
| supply#210 | supply#230 | supply#260 | supply#51  | supply#510 |
| supply#52  | supply#53  | supply#530 | supply#560 | e@25 #600  |

..warning..: resource no: #600 was not previously defined in MASTER PLAN
...........  range limit data:  45:50
.....................................................................
..........master plan no:  1984   scenario no:  4 ............

...B O U N D S..(VARIABLES WITH LIMITS)...

  (#10)=0:30             (#100)=900:900         (#20)=0:30
  (#410)=0:80            (#411)=40:40           (#430)=0:80
  (#431)=50:50           (#460)=0:80            (#461)=50:50
  (#610)=0:25            (#611)=15:15           (#630)=0:25
  (#631)=15:15           (#660)=0:25            (#661)=12:12
  (#710)=33:33           (#730)=33:33           (#760)=24:24
  (#600)=45:50
.....................................................................
```

```
 ..........master plan no:  1984   scenario no:  4 ............

ROW:    max: profit   <Maximize>
   - 25000*(#100) - 1*(#11) - 1*(#21) + 12000*(#51) + 22000*(#52)
   + 22000*(#53)

ROW:   demand#10      <EQ>  RHS= 0
   + 1*(#10) - .0333333*(@10 #100) - .0333333*(@11 #100)
   - .0333333*(@12 #100)

ROW:   demand#100     <EQ>  RHS= 0
   + 1*(#100) - 1*(@10 #100) - 1*(@11 #100) - 1*(@12 #100)

ROW:   demand#11      <EQ>  RHS= 0
   + 1*(#11) - 2000*(@10 #100) - 2000*(@11 #100) - 2000*(@12 #100)

ROW:   demand#20      <EQ>  RHS= 0
   + 1*(#20) - .166667*(@20 #310) - .166667*(@21 #330)
   - .192307*(@22 #360)

ROW:   demand#21      <EQ>  RHS= 0
   + 1*(#21) - 600*(@20 #310) - 600*(@21 #330) - 600*(@22 #360)

ROW:   demand#310     <EQ>  RHS= 0
   + 1*(#310) - 1*(@20 #310)

ROW:   demand#330     <EQ>  RHS= 0
   + 1*(#330) - 1*(@21 #330)

ROW:   demand#360     <EQ>  RHS= 0
   + 1*(#360) - 1*(@22 #360)

ROW:   e@15 #410      <EQ>  RHS= 0
   + 1*(#210) - 1*(#310) - 1*(#410) + 1*(#411)

ROW:   e@15 #430      <EQ>  RHS= 0
   + 1*(#230) - 1*(#330) - 1*(#430) + 1*(#431)

ROW:   e@15 #460      <EQ>  RHS= 0
   + 1*(#260) - 1*(#360) - 1*(#460) + 1*(#461)

ROW:   e@25 #610      <EQ>  RHS= 0
   + 1*(#510) - 1*(#610) + 1*(#611) - 1*(#710)
```

```
ROW:   e@25 #630      <EQ>  RHS= 0
   + 1*(#530) - 1*(#630) + 1*(#631) - 1*(#730)

ROW:   e@25 #660      <EQ>  RHS= 0
   + 1*(#560) - 1*(#660) + 1*(#661) - 1*(#760)

ROW:   supply#210     <EQ>  RHS= 0
   + 1*(#210) - .1*(@10 #100) - .1*(@11 #100)

ROW:   supply#230     <EQ>  RHS= 0
   + 1*(#230) - .1*(@10 #100) - .1*(@12 #100)

ROW:   supply#260     <EQ>  RHS= 0
   + 1*(#260) - .1*(@11 #100) - .1*(@12 #100)

ROW:   supply#51      <EQ>  RHS= 0
   + 1*(#51) - .04*(@10 #100) - .04*(@11 #100) - .04*(@12 #100)

ROW:   supply#510     <EQ>  RHS= 0
   + 1*(#510) - .62*(@20 #310)

ROW:   supply#52      <EQ>  RHS= 0
   + 1*(#52) - .1*(@10 #100) - .1*(@11 #100) - .1*(@12 #100)

ROW:   supply#53      <EQ>  RHS= 0
   + 1*(#53) - .38*(@20 #310) - .55*(@21 #330) - .61*(@22 #360)

ROW:   supply#530     <EQ>  RHS= 0
   + 1*(#530) - .45*(@21 #330)

ROW:   supply#560     <EQ>  RHS= 0
   + 1*(#560) - .39*(@22 #360)

ROW:   e@25 #600      <EQ>  RHS= 0
   - 1*(#610) - 1*(#630) - 1*(#660) + 1*(#600)
.....................................................................
```

Mathematical Formulation and Equation Method

DATA-ENTRY

MATHEMATICAL Formulations (EQUATIONS) Method

BEST-PLAN also supports the typical academic use of linear programming by formulating a number of linear equations, called ROWS/EQUATIONs (constraints), with VARIABLES which may contribute to a desired OBJECTIVE FUNCTION.

MODEL entry is accomplished through data statements similar to those described in the STEP-BY-STEP Flow Representation Method. The equation VARIABLES are treated and represented as RESOURCES(see Resource Description). The following data statements will therefore apply:

- RESOURCE Description data statements, using the keyword VARIABLES instead of RESOURCES.

- LIMITS data statements, to specify value constraints for a given VARIABLE. Consequently, the following data statements also apply: FIXED, L.L. and U.L.(see FIXED, L.L. and U.L. data entry statements)

Additional data statements are available to complete the MODEL entry. These will be discussed in this section with the given Demonstration Problem. The additional data statements are as follows:

- EQUATE/EQUATION data statements which establish the ROWS or CONSTRAINTS for the model.

- MAXIMIZE/MINIMIZE data statements which establish the model's OBJECTIVE FUNCTION.

BEST-PLAN by default, automatically creates an OBJECTIVE FUNCTION called " max: profit". This objective function is the COST ROW or the summation of all resources'(VARIABLES) PRICES. For a similar objective function requirement, the PRICE/COST data statement may be usedto automatically formulate this OBJECTIVE FUNCTION.

DEMONSTRATION PROBLEM: A.B.C. Distributing Company. Markets a product produced in three(3) location and sold in four(4) distributing outlets

The Distribution problem is as follows: The product is manufactured and distributed in the following locations:

SUPPLY	Units	DEMAND	Units
FLINT	150	CHICAGO	90
JAMESVILLE	40	CLEVELAND	70
ST.LOUIS	80	DAYTON	50
		MINNEAPOLIS	60
	270		270

The shipping costs per unit:

TO / FROM	CHICAGO	CLEVELAND	DAYTON	MINNEAPOLIS
FLINT	$27.00	$23.00	$31.00	$69.00
JAMESVILLE	10.00	45.00	40.00	32.00
ST.LOUIS	30.00	54.00	35.00	57.00

This problem can be formulated as follows:

```
#i0j  =no. of units shipped from source(i) to destination(j)
Cij   =shipping cost per unit for shipment(@i #j)
Si    =product supply at source
Dj    =product demand at destination
 i    =(1,2,3)   the supply points
 j    =(1,2,3,4) the destinations
```

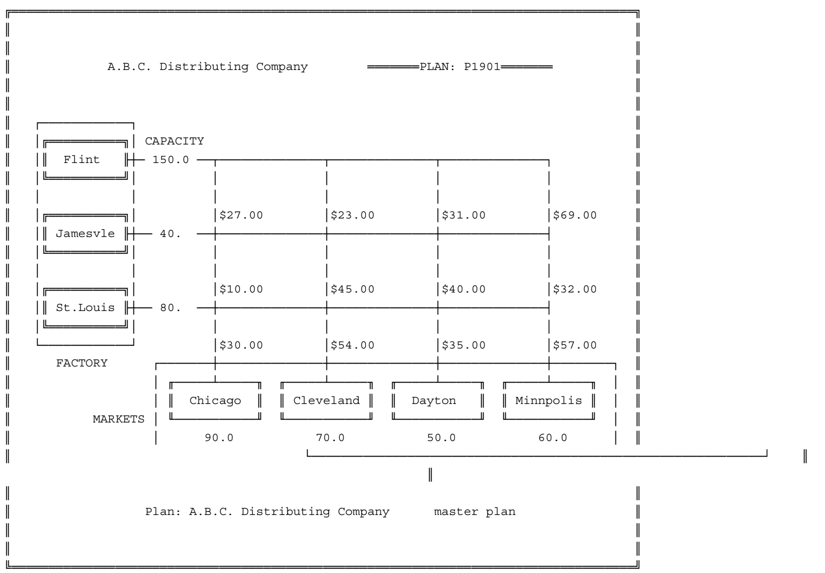
ANALYSIS SCENARIOS:

1. Minimize the cost for distributing the products.

2. What-if the freight from FLINT to DAYTON and MINNEAPOLIS is not available. What will be the distributing cost? Will the demands be satisfied?

DEMONSTRATION PROBLEM: A.B.C. Distributing Company. Markets a product produced in three(3) location and sold in four(4) distributing outlets

PROBLEM DIAGRAM



The shipping costs per unit:

TO / FROM	CHICAGO	CLEVELAND	DAYTON	MINNEAPOLIS
FLINT	$27.00	$23.00	$31.00	$69.00
JAMESVILLE	10.00	45.00	40.00	32.00
ST.LOUIS	30.00	54.00	35.00	57.00

DEMONSTRATION PROBLEM: A.B.C. Distributing Company. Markets a product produced in three(3) location and sold in four(4) distributing outlets

the SUPPLY EQUATIONS (Si) can be expressed as

```
#101 + #102 + #103 + #104 = S1 = 150          (1)

#201 + #202 + #203 + #204 = S2 =  40          (2)

#301 + #302 + #303 + #304 = S3 =  80          (3)
```

the DEMAND EQUATIONS (Dj) can be expressed as

```
#101 + #201 + #301 = D1 =  90          (4)

#102 + #202 + #302 = D2 =  70          (5)

#103 + #203 + #303 = D3 =  50          (6)

#104 + #204 + #304 = D4 =  60          (7)
```

the OBJECTIVE FUNCTION will be the shipping COST multiplied by the corresponding volumes: Cij * (#i0j).

```
MINIMUM = 27.0* #101 + 23.0* #102 + 31.0* #103....
 ...........+ 35.0* #303 + 57.0* #304
or.

MAXIMUM = -27.0* #101 - 23.0* #102 - 31.0* #103....
............- 35.0* #303 - 57.0* #304
```

The second form of the OBJECTIVE FUNCTION will allow us to use the default automatically created by BEST-PLAN labeled " max:profit". Shipping COSTS can then be entered using COST data statements.

VARIABLES IDENTIFICATION

VARIABLES will be treated just like RESOURCES as discussed in the preceding section. They will carry the same informational data as entered by the planner during data-entry and same data computed by BEST-PLAN after the optimization of the plan. The minimal informational requirement for a variable is the variable number (1-9999) and name identification.

Variable data declaration is as follows:

```
VARIABLE #nnnn = name identification(30-characters)

        #nnnn = variable number (1-9999)
```

VARIABLE,RESOURCE are synonymous and can be used interchangeably.(see. UTILITY)

Additional input information for a variable are as follows:

- Price or cost — Cost is a negative price. The data apply to variables contributing to the default OBJECTIVE FUNCTION " max: profit" when it is used.

- Limits(low/upper) — Planning limits control how much the variable will vary for the model. All variables default to a no-limit(free) assignment, which means that any positive value is acceptable.

Solution data are computed by BEST-PLAN after the model goes thru the optimization. A variable will have the following solution data:

- Volume. This amount shows the variable contribution to the optimal objective function. This volume is controlled by any possible low and/or upper limit imposed by the planner.

- Internal Cost. This data element will not be valid if the model is a PURE Mathematical Formulation. Cost propagation relates more to the STEP-by-STEP Flow Representation Method.

- Total price. This is the overall value of the variable, which is the volume multiplied by the unit price or internal cost depending on whether the cost is supplied or computed.

- Reduced Cost. Also known as shadow price or INCENTIVE. If the variable has reached its upper limit, as imposed by the planner, the incentive price shows how much the optimal objective function could be increased or decreased by increasing or decreasing the upper limit for this variable by one (1.0) unit.

The system uses the following naming convention for the data mentioned:

	RAW SOLUTION LIST
code . . . . #nnnn =variable no.	#nnnn
name identification (assigned by planner)	NM
price or cost(unit)	P$
* internal cost	I$
low limit	LL
upper limit	UL
fixed limit(low=upper)	FX
volume	VL
total price	T$
reduce cost(shadow price)	R$

NOTE: The product of the cost propagation program (internal cost) will not be valid for a model developed using a PURE Mathematical Formulation Method, since it relates more to the STEP-by-STEP Flow Representation Method.

PRICES AND LIMITS DATA STATEMENTS

The planner can enter prices and limits for a number of variables together in one data statement.

PRICE/COST

The PRICE data statement is intended for variables contributing to the default objective function. This objective function is labeled " max: profit".

```
      PRICES #nnnn = value, #nnnn = value
                  .........#nnnn = value;

             nnnn = variable numbers (1-9999)
            value = unit price or cost
                    (positive) sold
                    (negative) bought
```

(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator

PRICES,VALUES are synonymous and can be used interchangeably.(see. UTILITY)

COSTS,EXPENSES are synonymous and can be used to replace PRICE if the values are negative. This will reverse the sign (positive to negative) internally.

```
'.........................................................
'                        prices
'.........................................................

  PRICES  #101=-27.0       '.shipping FLINT-CHICAGO
          #102=-23.0       '.shipping FLINT-CLEVELND
          #103=-31.0       '.shipping FLINT-DAYTON

          #301=-30.0       '.shipping ST.LOUIS-CHICAGO
          #302=-54,0       '.shipping ST.LOUIS-CLEVELND
          #303=-35.0;      '.shipping ST.LOUIS-DAYTON

'.........................................................
```

PRICE/COST

Variable prices and name identifications can be combined together using the PRICES and COSTS data statements. Their corresponding order is immaterial.

```
PRICES #nnnn /name identification=value
       #nnnn =name identification/value
       #nnnn =value/name identification
       #nnnn /value=name identification;

        nnnn = variable numbers (1-9999)
        name identification (30-characters)
       value = unit price or cost
```

```
(/) slashes and (=) equals are used as separators
(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator
```

```
'.........................................................
'             prices
'.........................................................

  PRICES  #101=-27.0/shipping FLINT-CHICAGO
          #102/-23.0=shipping FLINT-CLEVELND
          #103/shipping FLINT-DAYTON    =-31.0

          #301=-30.0/shipping ST.LOUIS-CHICAGO
          #302=-54,0/shipping ST.LOUIS-CLEVELND
          #303=-35.0/shipping ST.LOUIS-DAYTON;

'.........................................................
```

LIMITS

The planner specifies value limitations for the variables through the LIMITS data statements.

```
LIMITS #nnnn = value1:value2,#nnnn = value1:value2
              .............#nnnn = value1:value2;

       nnnn = variable numbers (1-9999)
     value1 = low limit
     value2 = upper limit
```

(:)colons are used to separate low& upper limits
(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator

LIMITS,RANGE,BOUNDS are synonymous and can be used interchangeably.(see. UTILITY)

```
'.........................................................
'          demands on shipments
'.........................................................
'
  LIMITS  #101=10. : 20.  'shipment FLINT-CHICAGO
          #201=10. : 15.  'shipment JMVILLE-CHICAGO
          #301= 5. : 10.; 'shipment ST.LOUIS-CHICAGO

'.........................................................
```

LIMITS

Variable limits and name identifications can be combined together using the LIMITS data statements. Their corresponding order is immaterial.

```
LIMITS #nnnn /name identification=value1:value2
       #nnnn =name identification/value:value2
       #nnnn =value1:value2/name identification
       #nnnn /value1:value2=name identification;

        nnnn = variable numbers (1-9999)
        name identification (30-characters)
      value1 = low limit
      value2 = upper limit

(/) slashes and (=) equals are used as separators
(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator
```

```
'........................................................
'        demands on shipments
'........................................................
'
  LIMITS  #101=10. : 20./shipment FLINT-CHICAGO
          #201/10. : 15.=shipment JMVILLE-CHICAGO
          #301/shipment ST.LOUIS-CHICAGO    =5. : 10.;

'........................................................
```

FIXED

Both lower and upper limits may be the same value for certain variables. The planner may want a very tight control on production. The FIXED data statement may be used.

```
FIXED #nnnn = value, #nnnn = value
           .........#nnnn = value;

     nnnn = variable numbers (1-9999)
    value = both low and upper limit
```

(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator

FIXED,HOLD,SET are synonymous and can be used interchangeably.(see. UTILITY)

```
'.........................................................
'       set shipment between FLINT-DAYTON=0.0
'       set shipment between FLINT-MINNEAPOLIS
'.........................................................

    FIXED  #103=0.0       '.shipment FLINT-DAYTON
           #104=0.0;      '.shipment FLINI-MINNEAPOLIS

'.........................................................
```

FIXED

Variable FIXED limits and name identifications can be combined together using the FIXED data statements. Their corresponding order is immaterial.

```
      FIXED #nnnn /name identification=value
           #nnnn =name identification/value
           #nnnn =value/name identification
           #nnnn /value=name identification;

            nnnn = variable numbers (1-9999)
            name identification (30-characters)
           value = low and upper limit

     (/) slashes and (=) equals are used as separators
     (,) commas and (next-line) are used as separators
     (;) semicolon is used as terminator
```

```
   '.......................................................
   '       set shipment between FLINT-DAYTON=0.0
   '       set shipment between FLINT-MINNEAPOLIS
   '.......................................................

      FIXED  #103/shipment FLINT-DAYTON      =0.0
             #104/shipment FLINI-MINNEAPOLIS =0.0;

   '.......................................................
```

L.L.
U.L.

The lower and upper limits can be entered separately using the L.L. or the U.L. data statements respectively.

```
L.L. #nnnn = value, #nnnn = value
          .........#nnnn = value;

U.L. #nnnn = value, #nnnn = value
          .........#nnnn = value;

     nnnn = variable numbers (1-9999)
    value = low or upper limit

(,) commas and (next-line) are used as separators
(;) semicolon is used as terminator
```

L.L.,LOW,MIN,MINIMUM are synonymous and can be used interchangeably.

U.L.,UPPER,MAX,MAXIMUM are synonymous and can be used interchangeably.(see. UTILITY)

```
'.......................................................
'         demands on shipments
'.......................................................
'
  L.L.  #101=10.     'shipment FLINT-CHICAGO
        #201=10.     'shipment JMVILLE-CHICAGO
        #301= 5.;    'shipment ST.LOUIS-CHICAGO

'.......................................................
```