DS Lab

Decision Support Laboratory

"The Visual Spreadsheet"

The best way to "quick start" with DS Lab is to view the demonstration included on the accompanying diskette. DS Lab is very easy to use, but its innovative approach to modeling requires a minimum of introduction. The demonstration will show you how DS Lab works in about 15 minutes.

After that, explore DS Lab:

If you like to "dive" into new applications by trying the various commands, use the DS Lab help command. The section titled "Description of the Work Environment" will give you a detailed explanation of the menus and toolbars.

If you want to know more about DS Lab first, read the various sections of the help file.

At the beginning, we recommend you keep the help file open and use Alt+Tab to switch between help and DS Lab as you explore the software. To learn more about Alt+Tab, please look up "Switching" in the index of your Windows manual.

There are several sample files which show you what a finished model looks like. An index to these files is listed below.

If you wish to work through a tutorial of DS Lab, the first section of this "Quick Start" manual will guide you, step by step, through building a model.

Enjoy DS Lab.

The "Quick Start" manual

This manual contains the following sections:

1. DS Lab Tutorial

The purpose of this tutorial is to introduce you to the main characteristics of DS Lab. You will learn to place elements in the model, to connect them and to write formulas. There is also a brief introduction to exporting data from DS Lab to other Windows applications for presentation purposes.

Let's illustrate the main characteristics of DS Lab by building a simple budget model which will calculate the main elements of a business budget:
> Gross Revenue
> Operating Costs
> Profit Margin
> Net Profit
> Cumulative Net Profit

The functions and operations used in this example will be explained in greater detail in the help file.

1.1 Let's Get Started

DS Lab is a Windows application, so it is launched from the standard Windows interface, Program Manager.

The DS Lab installation program created a Program Group called DS Lab. To open the group, double-click on the group icon.
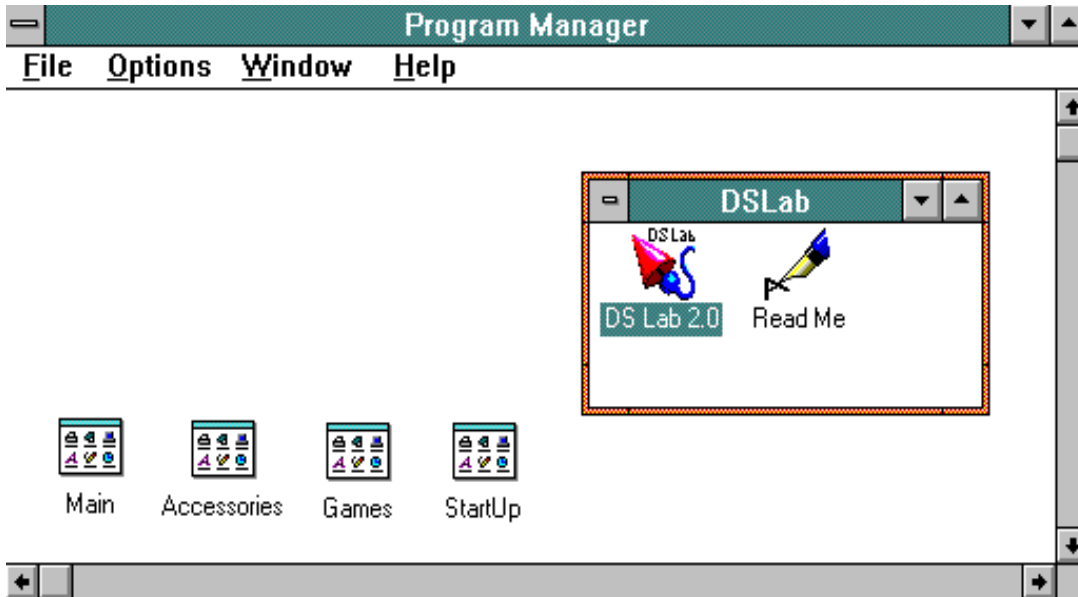
To start DS Lab, double-click on the DS Lab icon or select the icon and press Enter.
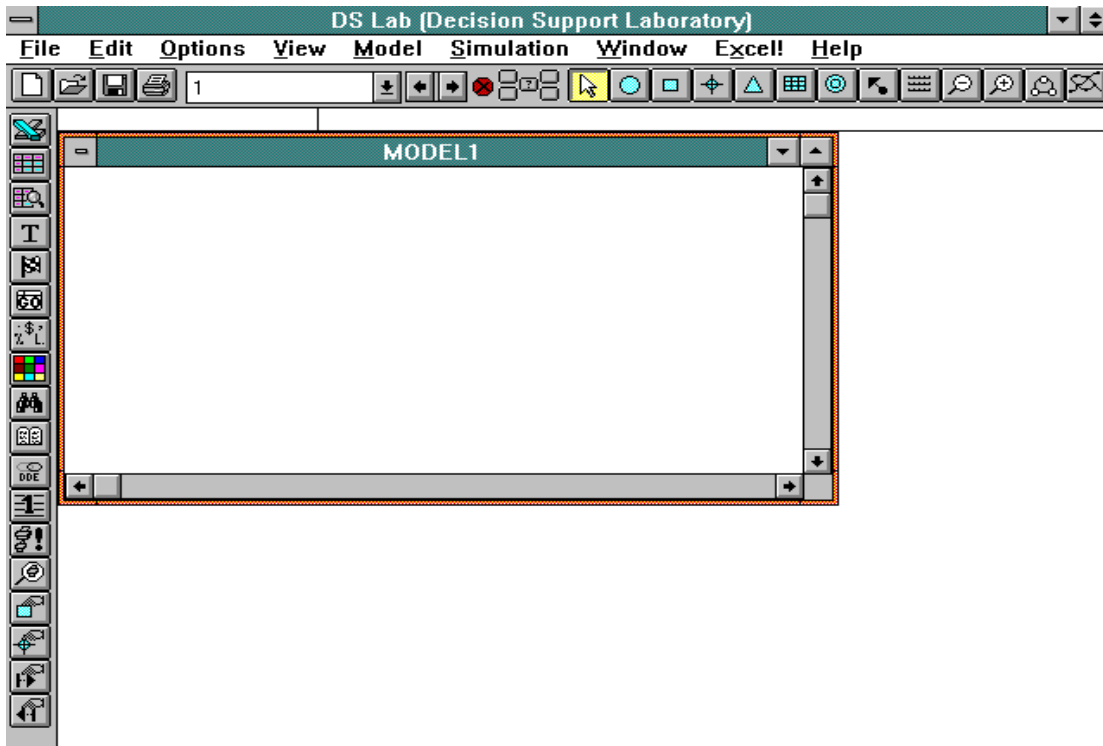
At start-up, DS Lab shows an empty model window named MODEL1, in which the budget model will be built. After you save the model, the name you give it will appear on the title bar.

1.2 Building the Model

In DS Lab, each element of the model is graphically represented by a symbol. To enter an element, click the tool representing the element (located on the horizontal toolbar). Or from the Model menu, choose the type of element required.
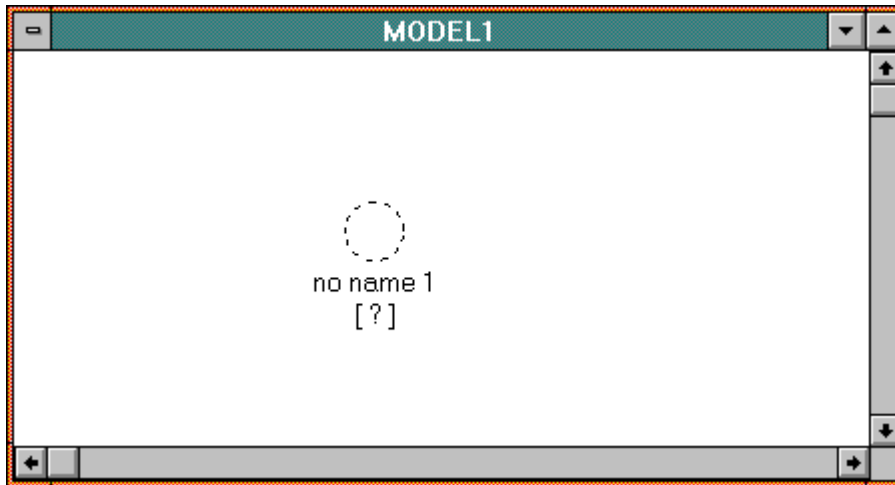
To place a variable element in the model:
1.Click the Variable tool

[variable tool icon]

representing an element whose value is defined by a formula. The cursor assumes the shape of the element to be inserted.

2.   Move the pointer into the model window.
3.   Click the mouse button.

DS Lab has entered a variable, calling it "no name 1". Let's give this variable a meaningful name. Under the toolbar we can see the name of the variable. This space is called the Edit Bar because it is used to edit the names of elements.

To change the element's name:
1.   Click in the Edit Bar (or press F2).
2.   Backspace to delete the current name.
3.   Type Gross Revenue.
4.   Press ENTER.
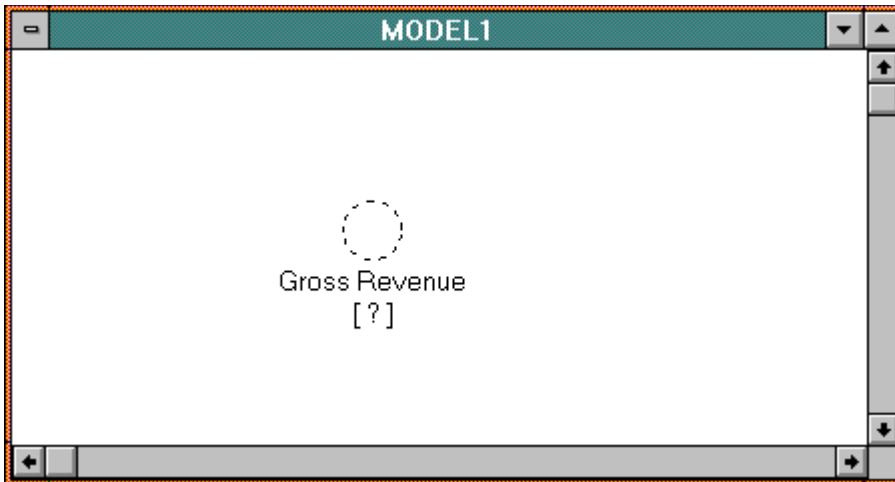
Next we will create an input element called Unit Price and a series element called Sales Volume.

To create an Input element:
1.   Click on the Input tool.

2.   Click in the model window.
3.   Name the new element Unit Price.

To create a Series element:
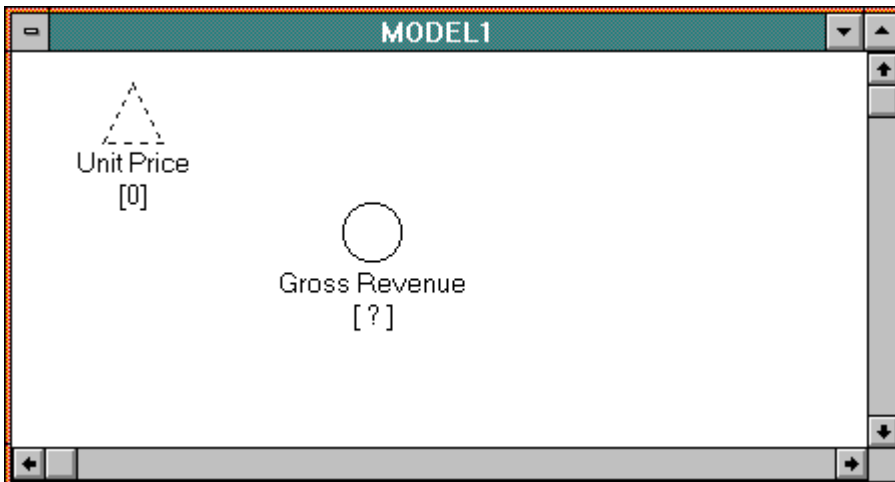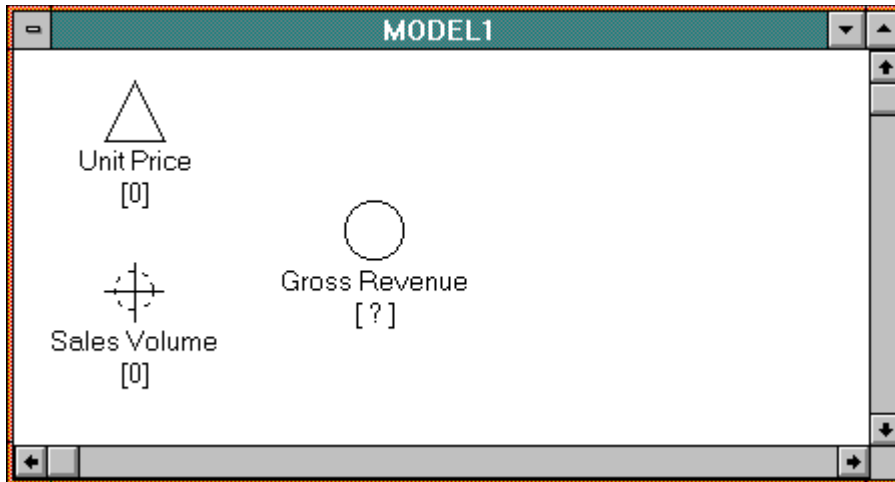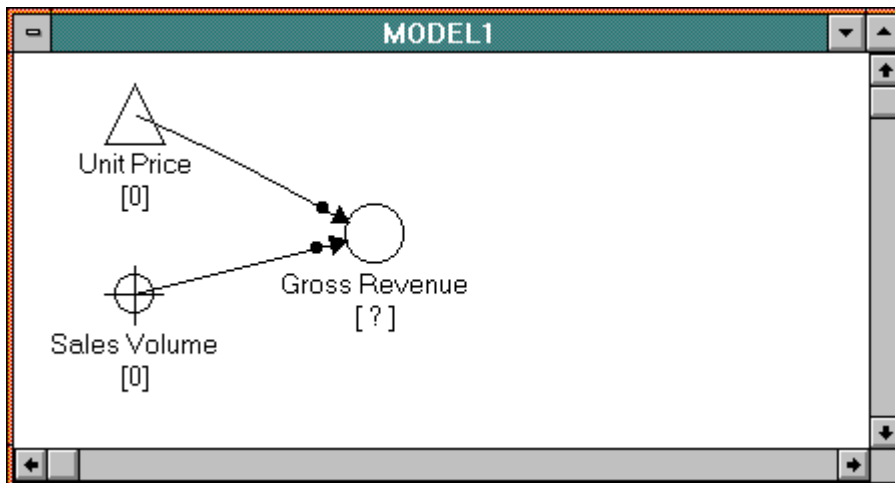1.   Click on the Series tool.

2.   Click in the model window.
3.   Name the new element Sales Volume.

The next step in building our model is to connect the three elements as shown below.

To connect the elements:
1.   Click the Connect tool.

2.   Position the pointer over the Unit Price element (the start point).
3.   Click and hold the left mouse button while moving the pointer to the Gross Revenue element (the destination).
4.   Release the mouse button.

Repeat this process with Sales Volume as the start point and Gross Revenue as the destination. The elements can easily be moved around in the model at any time so as to give it a logical structure. The arrows can also be selected, by clicking on the dot, and then curved with the mouse. You can also zoom in or out to display a larger or smaller portion of the model. Elements can be copied, cut and pasted, etc.

DS Lab is built with an interpreter that enables it to understand the logical connections between the elements in the model and to evaluate the variables and their scripts.

In DS Lab the connection arrows are not just illustrative, but represent information available to the formulas defining the variables: the value of the destination variable is determined by that of the

element(s) connected to it. In our example, Gross Revenue will be calculated by estimating the monthly Sales Volume and multiplying it by the Unit Price.

The next stage in building our model is to make a copy of the Gross Revenue variable and use it again in another location.

To make a Shadow element:
1.   Click on the Shadow tool.
[Shadow tool icon]

2.   Click on the Gross Revenue element.
3.   Position the pointer where you want to place the shadow in the model and click.
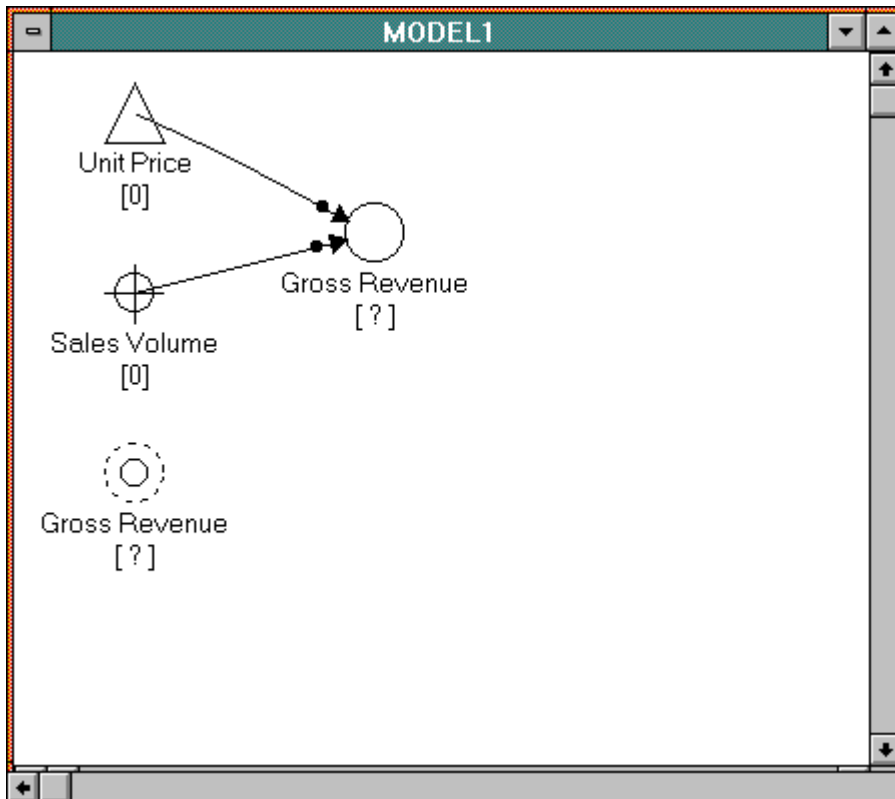
Next we will create an element called Operating Costs, which will be calculated as a percent of Gross Revenue.
Using the technique described above, create a Variable element named Operating Costs.

To make a Constant element:
1.   Click on the Constant tool.
[Constant tool icon]

2.   Click in the model window.
3.   Name the new element Percentage Factor.

Connect Gross Revenue and Percentage Factor to Operating Costs, using the Connect tool as described above.
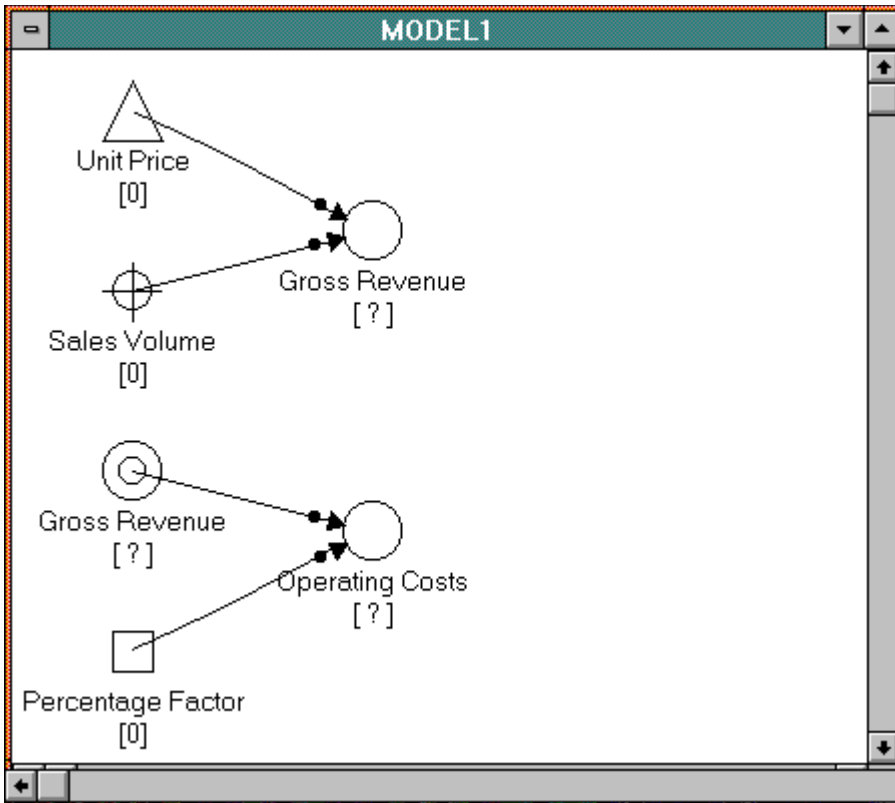
Now enter the last four elements: Profit Margin (variable), Other Costs (series), Net Profit (variable) and Cumulated Net Profit (variable) then connect them as shown in the following diagram:
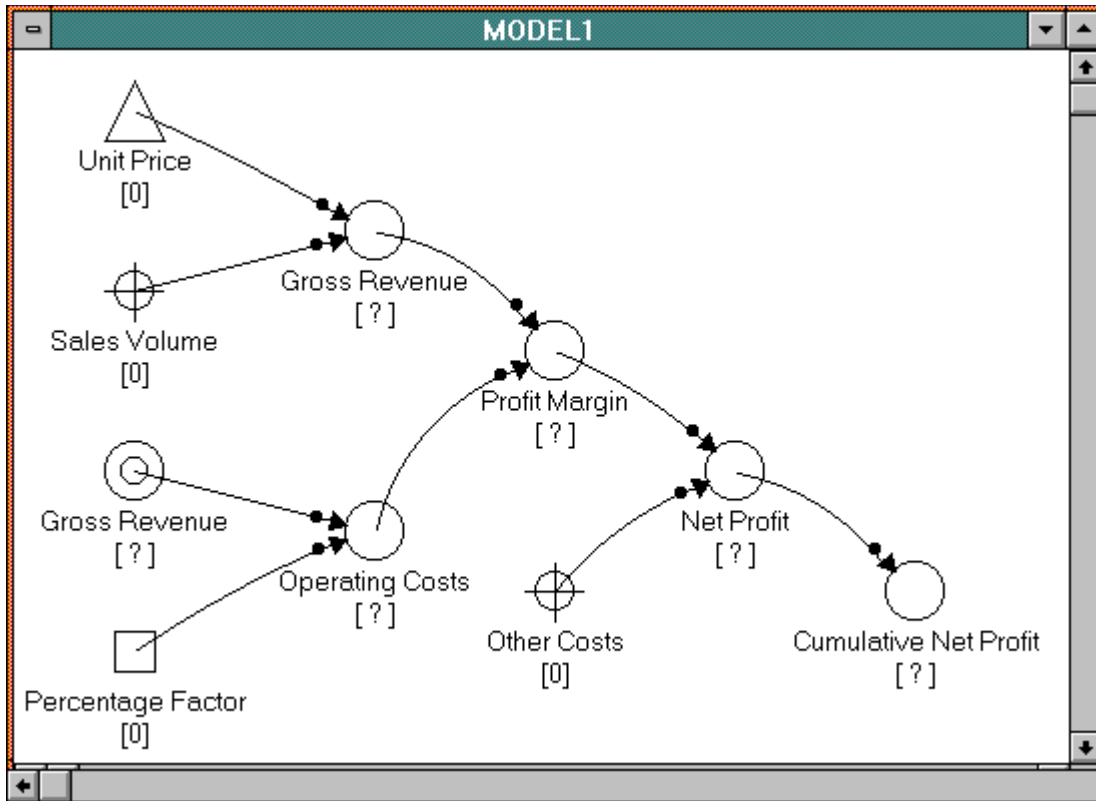
1.3 Defining the Mathematical Values

Now that the graphical and relational aspects of the model are complete, we need to enter the formulas and instructions needed to calculate the variables in the simulation. These make up what from now on will be called the Script of each variable.

The script is entered in a dialog box which is accessed by double-clicking on the variable element. For example, if you double-click on the Gross Revenue element you will see the following window:

```
┌─────────────────────────────────────────────────────────────────────┐
│                              Variable                                 │
├─────────────────────────────────────────────────────────────────────┤
│ Connected Elements      Functions/Instructions        Name           │
│ Sales Volume            Abs( number )              ▲  Gross Revenue   │
│ Unit Price              AddDay( internal_date , days )                │
│                         AddMonth( internal_date , mon  ☐ Self-Reference  [ [ & < │
│                         AddWeek( internal_date , weel                 │
│                         AddYear( internal_date , years Starting Value  ] ] | > │
│                         ADUE                                          │
│                         AFVFutureValueIncreasingCash▼  0              + ▪ ^ ! │
│ Script                                             ⊠ Always Calculate  - / % = │
│ ┌─────────────────────────────────────────────────────────────────┐ ▲│
│ │                                                                   │  │
│ │                                                                   │  │
│ │                                                                   │  │
│ │                                                                   │ ▼│
│ DDE:┌──────────────────────────────────────────┐  Paste Link  Update │
│     └──────────────────────────────────────────┘                     │
│ ┌──────────────┐                   ┌──────────┐  ┌──────────┐         │
│ │ Edit Script  │                   │   OK     │  │  Cancel  │         │
│ └──────────────┘                   └──────────┘  └──────────┘         │
└─────────────────────────────────────────────────────────────────────┘
```

To enter the Script:

1. Position the pointer in the Script text box.
2. Click the mouse button.
3. From the Connected Elements list box, choose (double-click) Sales Volume.
4. Click the * symbol on the calculator-style keypad on the right, or press the * key.
5. From the Connected Elements list box, choose (double-click) Unit Price.

OR: Instead of steps 3-5, you may directly type in the formula:

    Sales Volume * Unit Price

6. Choose the OK button (or press Enter).

Now enter the other scripts in the same way:

    For Operating Costs, the formula is:
    Gross Revenue * Percentage Factor
    For Profit Margin, the formula is:
    Gross Revenue - Operating Costs
    For Net Profit, the formula is:
    Profit Margin - Other Costs
    For Cumulative Net Profit, the formula uses Self-Reference, that is, the variable's name is
used in its own formula. This is done by selecting the Self-Reference check box. This causes the
variable's name to be inserted in the list of Connected Elements which may be used in the script.
The formula is:
    Cumulative Net Profit + Profit Margin

The Percentage Factor and Unit Price elements are treated differently. As an Input and a Constant, respectively, all we have to do is give them the value that they are to assume for the first simulation.

To do this, double-click on each element to open its dialog box and enter a value in the Value box: 0.55 for Percentage Factor and 100 for Unit Price. At each recalculation we will be asked to confirm the value given to Unit Price, while Percentage Factor will remain the same until we decide to change it.

1.4.   Setting Parameters for the Simulation

The step units of the current model can be defined in the dialog box called Simulation Parameters (accessed by choosing Parameters from the Simulation menu) or clicking the Simulation Parameters tool.

Our example is a budget model for the year 1993 with monthly steps, so we choose the Step Unit Month. This means that the formulas we have given will be recalculated for each month. The First Step is defined as January 1993 and the Number of Steps which we want to simulate as 12. The Last Step will automatically become December 1993.

The value shown below each element in the model refers to the Current Step, that is, the step most recently calculated. At the beginning, this is the First Step, and then assumes all the intermediate values in turn until reaching the Last Step.

The last parameter, Simulation Length, allows you to run the simulation for a number of steps which is less than the number in the model (defined by Number of Steps). This feature enables you to evaluate interim results and, if you wish, to change some of the values before continuing the simulation. For this example, enter 6 in the Simulation Length box, instructing the model to calculate only six of the twelve steps.

Simulation Parameters

Step Unit	Month ⬇
Multiple Unit	1

First Step	N° of Steps	Last Step
January 1993	12	December 1993

Current Step	January 1993 ⬇
Simulation Start	January 1993 ⬇
Simulation Length	6

[User defined Unit] [OK] [Cancel]

1.5. Entering the values of the Series

Before starting the simulation, we still have to enter the values of the two Series: Sales Volume and Other Costs. This data can be entered either in the dialog box of the series or in a table, opened by choosing Edit Series from the Model menu or clicking the Edit Series tool.

The values can then be entered manually or imported from another application using the Copy and Paste commands or DDE links.

In our example we have to insert twelve values for each of the two series:

Other Costs	800	800	800	600	600	500	500	400	600	600
	600	600								

Sales Volume	10	12	15	18	23	30	27	10	17	22
	27	33								

Edit Series

Edit

800

	Starting Values	January 1993	February 1993	March 1993	April 1993
Other Costs	0	800	800	800	600
Sales Volume	0	10	12	15	18

The above figure also illustrates one of the main characteristics of DS Lab: the representation of the time factor. Each column of the series corresponds to a step in our simulation.

1.6. Starting the Simulation

We are now ready to start the simulation by choosing Recalculate from the Simulation menu or clicking the Recalculate tool.

A dialog box will appear confirming the input's value. Edit the value, then choose the OK button.

In the drop-down list box on the toolbar we see the Current Step run through from January 1993 to June 1993. The value under each element changes correspondingly.

The simulation stops at June 1993, since we set the Simulation Length to 6 (months). To

complete our simulation we must continue the Calculate operation by clicking the Calculate tool.

The Current Step will now go to December 1993.

Note that the Traffic Light on the toolbar to the right of the Current Step box has changed from red to green. This means that the values shown on the screen have been correctly calculated. If any part of the model is changed, the Traffic Light will turn red until the Recalculate command is given.
Numeric values may be shown in the format $ *value*; this and other formatting options can be set in the Options menu, while those concerning sizes, zoom levels and tools to move about the model are in the View menu. Each of these commands has a corresponding button on a toolbar.

Going back to our model, if we want to analyze the effect on Net Profit of changes in any of the inputs (Price, for example), we have two options:
Double-click on the input element, change its value, then repeat the simulation;
OR:
From the Simulation menu, choose Recalculate (or click the Recalculate tool).

1.7.   Reporting the Results
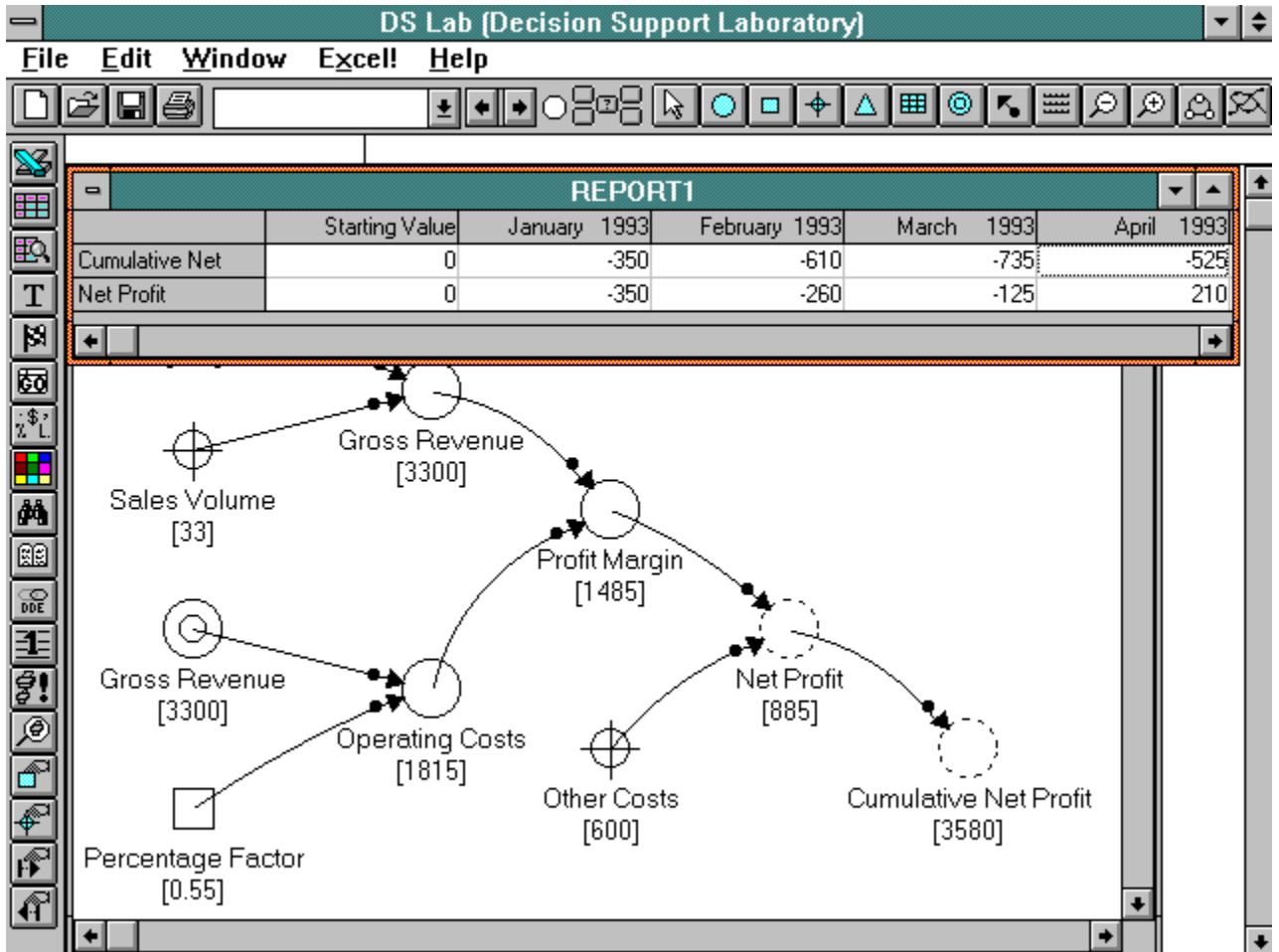There are several ways to compare the results of different simulations:
Internal report: select the elements you want to report (in our example Net Profit and Cumulative Net Profit) using the Selection Arrow tool

and clicking on the first element and then, while holding down the SHIFT key, clicking on the second element. Then from the Simulation menu, choose Report (or click the Report tool).

DS Lab (Decision Support Laboratory)

File   Edit   Window   Excel!   Help

REPORT1

	Starting Value	January 1993	February 1993	March 1993	April 1993
Cumulative Net	0	-350	-610	-735	-525
Net Profit	0	-350	-260	-125	210

Gross Revenue
[3300]

Sales Volume
[33]

Profit Margin
[1485]

Gross Revenue
[3300]

Net Profit
[885]

Operating Costs
[1815]

Other Costs
[600]

Cumulative Net Profit
[3580]

Percentage Factor
[0.55]

Excel spreadsheet: select the elements you want to report (for example, Net Profit and Cumulative Net Profit), then from the Simulation menu, choose Export to Excel (or click the Export to Excel tool).

Note: Excel must be already running or located on the PATH defined in your computer's AUTOEXEC.BAT file.

To automatically create a chart every time you export data to Excel, enter the Workspace Setup dialog box by choosing Workspace... from the Options menu and enable the Create Chart option by clicking its check box.

Workspace Setup

Excel
Version: English ▼

☒ Create Chart   ☒ Excel Menu

Toolbar
☒ Horizontal

☒ Vertical   [Default]   ☒ Edit Bar

Default Settings for Copy
◉ Steps in Columns   ☒ Step Headings
○ Steps in Rows   ☒ Element Names
                       ☒ Include Starting Values

[OK]   [Cancel]

DS Lab (Decision Support Laboratory)

File Edit Options View Model Simulation Window Excel! Help

December 1993

MODEL1

Microsoft Excel - Chart3

File Edit Gallery Chart Format Macro Window DS Lab Help

Legend

■ Cumulated Net Profit
■ Net Profit

Ready   NUM

Export the data to other Windows applications with the normal Copy and Paste procedure.

Link to an Excel spreadsheet: In this case any change in the model will automatically modify the values in the Excel spreadsheet. To make the connection, choose the Link with Excel command from the Simulation menu, select the elements to export, identify the destination spreadsheet and the top left corner of the table to be created (e.g.R1C2):

Note: Excel must be already running to make the connection.
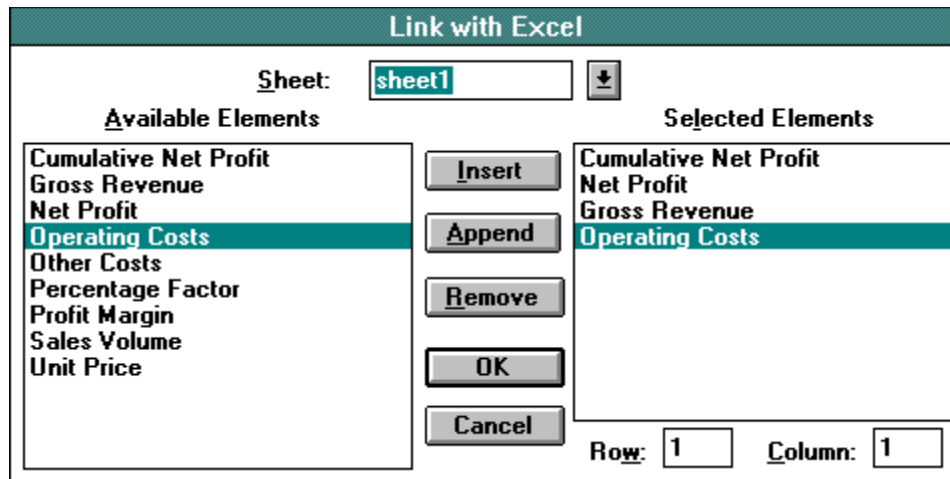
Hot DDE links (Single element) to another Windows application: Data exported this way will be automatically updated by DDE. The procedure is identical to a normal Copy and Paste, except that the command in the destination application is Paste Link instead of Paste.

This procedure allows you to connect only one element at a time, and the results obtained depend largely on the destination application. For example, a link with Excel 3.0 will export only one value (the first), unless before choosing Paste Link you select a zone of the worksheet (e.g. a row or a double row) large enough to contain all the values. With Excel 4.0 this is not necessary because of the particular way it reads data from the Clipboard.

To link several elements at one time, you are advised to use the Copy Link commands described below.

Hot DDE links (Multiple elements) to another Windows application: The Copy Link Current and Copy Link Total commands allow you to link several elements at one time. Data is exported into another Windows application by DDE, so it will be automatically updated. Copy Link Current exports the values of the current step only, whereas Copy Link Total exports the values for all steps.

The procedure is the same as a normal Copy and Paste, except that the DS Lab command is Copy Link Current or Copy Link Total, and the command in the destination application is a normal Paste (NOT Paste Link).

Note that the names of elements and simulation steps, though exported, are not included in the DDE links, so any changes in them will not be updated.

The above operations using DDE and Link with Excel allow you to leave all the reporting of output data to other Windows applications. Only the reporting of the model in graphic form and of the scripts must be done by DS Lab.

2. The manual addenda

2.1. Changes to the manual and help file

In Section 3.9.3. "*Error Values*" of Chapter 3 and Appendix C "*Error Values*" the following are error values not documented:

[**#EVERY**]   Error in the arguments of the Every function.
[**#EVERYDAY**]   Error in the arguments of the EveryDay function.
[**#EVERYMONTH**]   Error in the arguments of the EveryMonth function.
[**#EVERYWEEK**]   Error in the arguments of the EveryWeek function.
[**#EVERYYEAR**]   Error in the arguments of the EveryYear function.

In Chapter 6 the following sections have been modified:

6.48.   Every - Function
Description
Assumes the value 1 (**TRUE**) for steps separated by increment intervals, starting from a given step (start_step). Counts steps based on the **CURRENTSTEP** value if the Step Unit is **Unit**, counts steps based on the **SIMSTEP** if the Step Unit is anything other than **Unit**.
.
Syntax
return = Every(start_step, increment)
Returns
1 (**TRUE**) or 0 (**FALSE**).
The presence of errors in the arguments will cause the function to return the error value **#EVERY**.
Comments
Used to determine steps separated by fixed intervals. It is important to understand the distinction between CURRENTSTEP (the label for the current step) and SIMSTEP (the sequence of the current) to use this function effectively.

When the parameter Step Unit is set to **Unit**, the function refers to the number of the Current Step (**TIME** or **CURRENTSTEP** variables), and counts starting with the function argument start_step and the given interval.

For example, when the Step Unit is **Unit** the script Every(1, 3) has the value 1 (TRUE) when the variable **CURRENTSTEP** (that is the label displayed in the simulation step window) assumes the values 1, 4, 7, 10, etc. Which step this refers to in the sequence of steps will depend on the parameter **Starting step**. If the parameter **Starting step** is 5 (**CURRENTSTEP** 1,4 will not show), the third step **SIMSTEP** 3 (**CURRENTSTEP** value 7), will be the first step for which the condition is true and for each third step thereafter.

When the parameter Step Unit is set to a measure other than Unit, the function refers to the period number (**PERIOD** or **SIMSTEP** variables). It cannot use **CURRENTSTEP** to count because this is now contains a label (a month for example) rather than a number.

For example, when the Step Unit is set to **Month**, the script Every(1, 3) has the value 1(**TRUE**) when the variable **SIMSTEP** assumes the values 1, 4, 7, 10, etc. That is, beginning at the first step in the sequence, regardless of which month of the year it may be, and each third step thereafter. If the parameter **Starting Step** is February, Every(1,3) will be true for February, May, August etc.

6.49.   EveryDay - Function
Returns
1 (**TRUE**) or 0 (**FALSE**).
The presence of errors in the arguments will cause the function to return the error value

#EVERYDAY.

6.50.   EveryMonth - Function
Returns
1 (**TRUE**) or 0 (**FALSE**).
The presence of errors in the arguments will cause the function to return the error value
#EVERYMONTH.

6.51.   EveryWeek - Function
Returns
1 (**TRUE**) or 0 (**FALSE**).
The presence of errors in the arguments will cause the function to return the error value
#EVERYWEEK.

6.52.   EveryYear - Function
Returns
1 (**TRUE**) or 0 (**FALSE**).
The presence of errors in the arguments will cause the function to return the error value
#EVERYYEAR.

2.2. Printer Problems: HP Laserjet 4
As of this writing, the HP LaserJet 4 printer driver does not function properly with all applications. DS Lab is one such application. DS Lab graphics will print properly on your HP LaserJet 4 if you use the current HP LaserJet III driver supplied by HP or Microsoft. If you have doubts as to whether or not you have the latest printer driver, please contact HP.

3. The index of DS Lab sample files

3.1. Samples from the help file functions examples

*	AFV.LAB	Shows examples of the Advanced Future Value functions
*	APV.LAB	Shows examples of the Advanced Present Value functions
*	BOND.LAB	Shows examples of the Bond functions
*	CALCFROM.LAB	Shows examples of the CalcFrom function
*	FV.LAB	Shows examples of the Future Value functions
*	GETSTEP.LAB	Shows examples of the GetStep function
*	GOALSEEK.LAB	Shows examples of the LoopTime instruction
*	INSTR.LAB	Shows examples of the Script Language instructions
*	INVENTOR.LAB	Shows examples of the Inventory functions
*	MATH.LAB	Shows examples of the Mathematical and trigonometrical functions
*	POKE.LAB	Shows examples and explains the use of the Poke function
*	PV.LAB	Shows examples of the Present Value

functions
*	REQUEST.LAB	Shows examples and explains the use of the Request function
*	SP.LAB	Shows examples of the Stock Portfolio functions
*	ST.LAB	Shows examples of the Short Term Note functions
*	STAT.LAB	Shows examples of the Statistical functions
*	TREND.LAB	Shows examples of the three Trend functions

2.2. Samples Models

*	800PHONE.LAB	Shows comparison of 800 number calling plans
*	AFVINCCF.LAB	Shows an example of the AFVFutureValueIncreasingCashFlows function
*	BNDFNCT1.LAB	Shows an example of some of the Bond functions
*	CALLOPTN.LAB	Shows an example of the SPEquityCallOption function
*	CASHACNT.LAB	Shows an example of an account with interest calculation
*	FIN_STAT.LAB	Shows an example of financial statements
*	FXRTMRTG.LAB	Shows a fixed rate mortgage calculator
*	HM_BDGT.LAB	Shows an example of a home budget
*	INVENTRY.LAB	Shows an example of an inventory management model
*	MAILING.LAB	Shows an example of a mailing model
*	MORTCOMP.LAB	Compares values from FXRTMRTG.LAB and VRRTMRTG.LAB
*	MUNIBDGT.LAB	Shows a model that generates potential tender pricing for the re-purchase of outstanding municipal bonds
*	MUNISENS.LAB	Shows an example of sensitivity of the present value of cash flows from MUNIBDGT.LAB to changes in interest rates
*	PENSION.LAB	Shows an example of a pension plan disbursement decision
*	PV&IRR.LAB	Shows examples of the use of the Present Value and Advanced Present Value functions
*	REVMARG2.LAB	Shows an example of a Revenue and Gross Margin Model
*	TRNDFNCT.LAB	Shows an example of the Trend functions
*	VRRTMRTG.LAB	Shows a Variable Rate Mortgage calculator
*	YTDEXMPL.LAB	Shows examples of how to use scripts to calculate cummulative, year to date, annual, average and quarterly values.

4. Technical Support

If you have questions about DS Lab we can be reached **M-F 9:00 am - 5:00 pm Eastern time at (203) 861-1833**, on **CompuServe** our address is **71552,1645** or you can write to us at:

DS Group, Inc.
474 North Street
Greenwich, CT 06830-3449

5. DS Lab pricing

Thank you for trying DS lab. We wrote DS Lab with the goal of making it easier and more productive to develop quantitative models in Windows and to support you in your decision making.

We offer DS Lab (standard edition) for $195 ($295 after the introductory period), which allows the user to model up to 100 elements and 366 steps. We also include a printed 64-page Getting Started manual and 90 days of free telephone support. If you purchase DS Lab and choose to upgrade to DS Lab Pro within 90 days of purchase, we offer a 100% credit for your DS Lab purchase. After the 90 day period we offer a 10% discount on the price of DS Lab Pro for users of DS Lab.

For "industrial strength" decision modeling we offer DS Lab Pro for $995 ($1495 after the introductory period), which allows the user to model up to 1000 elements and 2048 steps. We also include a printed 550-page complete reference manual and lifetime free telephone support for current versions.