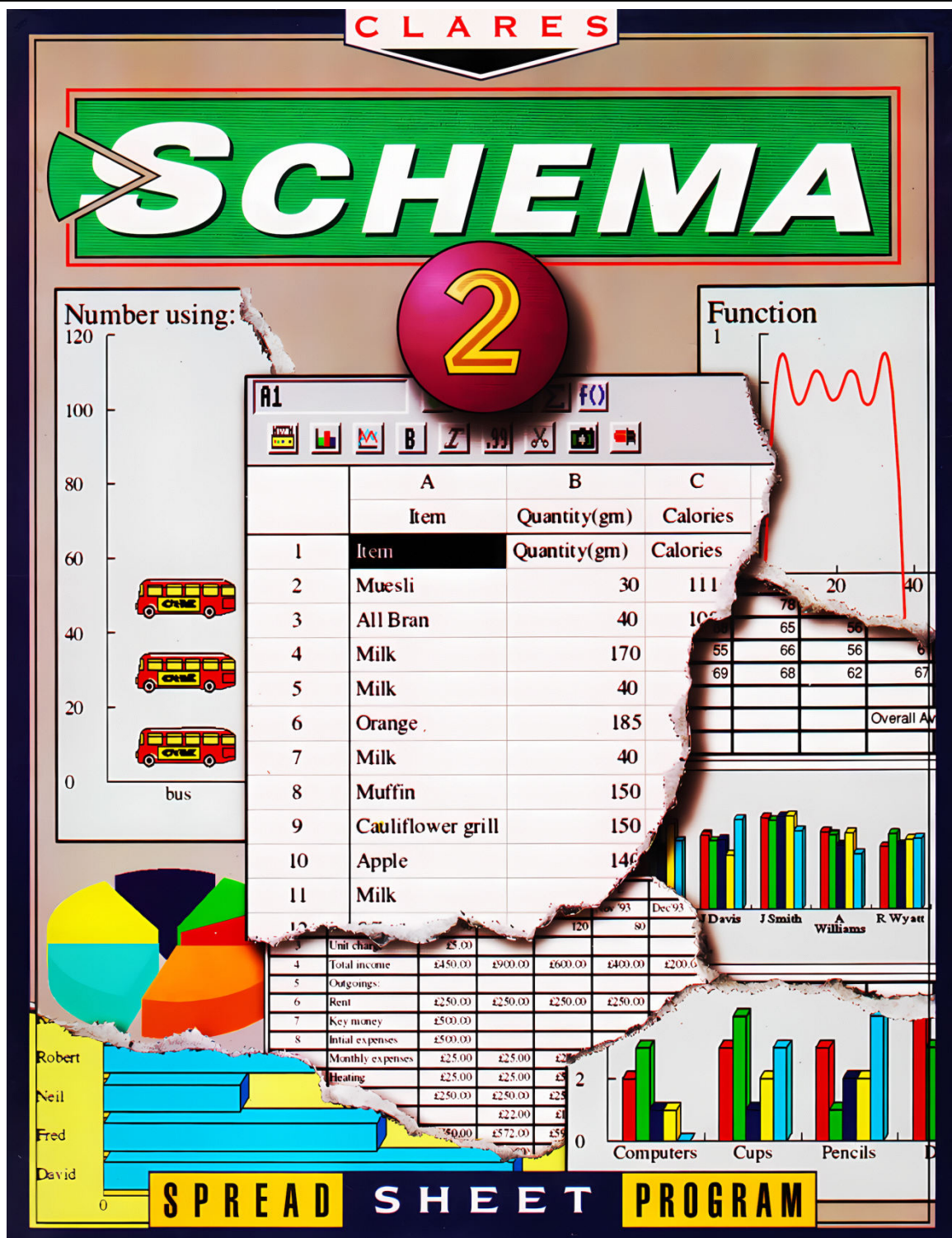




Schema2

User Guide

ProAction

3QD
DEVELOPMENTS LTD

APDL

Schema2

© CRM Ltd, ProAction & 3QD Developments Ltd 2026
All rights reserved.

No part of this product may be reproduced in whole or part by any means without written permission of the publisher. Unauthorised hiring, renting, lending, public performance or broadcasting of this product or its parts is prohibited.

While every care is taken, the publisher cannot be held responsible for any errors in this product, or for loss of data or consequential effects from the use of this package.

Software and Documentation supplied by
APDL
a part of
3QD Developments Ltd
Single User version downloaded from the free of charge software section of
www.riscos.com

Please note that any free of charge downloads from RISCOS.com are supplied without any technical support package as per the licence agreement.

This document last edited on 29/09/26



If you have found this application and this manual useful please consider making a donation to help cover the costs of cataloguing, documenting, updating and hosting the APDL applications. Scanning the QR code will take you to our donations page.

Contents

Introduction to Schema2 Macros	7	Example 7 - Biorythms	47
Typographic conventions	7	Ex 7.1 Dates, times and temporal values	47
		Ex 7.2 NOW()	48
		Ex 7.3 Other Standard Functions	48
		Summary	50
The Examples 9			
Example 1 - Class Records	11	Example 8 - Graphics	51
Ex 1.1 Starting up and opening a new sheet	11	Ex 8.1 The New Graph Dialogue	52
Ex 1.2 Entering text	14	Ex 8.2 Horizontal histograms	53
Ex 1.3 Correcting mistakes in spreadsheet cells	15	Ex 8.3 The Data Selection Dialogue	54
Ex 1.4 Filing or Saving a spreadsheet	15	Ex 8.4 The 3D effect	55
Ex 1.5 Closing a spreadsheet	16	Ex 8.5 Pie-charts	56
Ex 1.6 Ending a Schema2 session	17	Ex 8.6 Printing graphs	56
Ex 1.7 Reloading a spreadsheet	17	Ex 8.7 Exporting graphs	56
Ex 1.8 Saving the latest version of your spreadsheet	18	Summary	56
Ex 1.9 Selecting a block of cells	18		
Ex 1.10 Printing Spreadsheets	18	Example 9 - Decoding exam results	57
Ex 1.11 Using keyboard short cuts	19	Ex 9.1 Adding an extra row	57
Summary	19	Ex 9.2 Ranges	57
		Ex 9.3 Array Functions	58
Example 2 - How long is a piece of string?	20	Ex 9.4 Adding and deleting rows in the middle of ranges.	59
Ex 2.1 Setting up a sheet with a formula	21	Ex 9.5 Strings	59
Ex 2.2 Checking Formulae	21	Ex 9.6 Conditional Formulae	61
Ex 2.3 New Data	21	Ex 9.7 Hiding Rows or Columns.	63
Summary	21	Summary	63
Example 3 - Exam results	23	Example 10 - Financial predictions	64
Ex 3.1 Block Entries	24	Summary	65
Ex 3.3 Locking rows and columns	25		
Summary	26	Techniques 1 - Selecting blocks of cells	67
Example 4 - Sorting	27	Techniques 2 - Using the mouse to enter formulae	69
Ex 4.1 Sorting	27	Entering a function with the mouse	69
Summary	30	Using the Sum icon	69
		Entering a macro into a formula	69
Example 5 - Growing your savings	31	Entering a sheet variable into a formula	70
Ex 5.1 Changing the dimensions of a sheet	31	Entering cell and block names	70
Ex 5.2 Generating column headings	32	Terminating a formula with the Tick icon	70
Ex 5.3 Absolute References	32	Cancelling a formula with the Cross icon	70
Ex 5.4 The Go to cell dialogue box	34		
Ex 5.5 Number formats	34	Techniques 3 - Changing Column and Row Dimensions	71
Ex 5.6 Graphs	35	Choosing measurement units with the Choices Dialogue	71
Ex 5.7 Mixed references	36		
Summary	37	Techniques 4 - Changing sheet layout	73
		Using the Create New Spreadsheet dialogue.	73
Example 6 - Banners and Posters	38	Inserting rows	73
Ex 6.1 Effects	38	Deleting Rows	73
Ex 6.2 Column Widths and Row heights	39	Inserting columns	74
Ex 6.3 Big Cells	40	Deleting Columns	74
Ex 6.4 Images	41		
Ex 6.5 Scaling	42	Techniques 5 - Changing the margin content	75
Ex.6.6 Borders and Grid lines	42	Hiding and restoring System labels	75
Ex 6.7 Protection	44	Locking rows or columns	75
Summary	45	The Appearance of Row and Column Labels	75

Techniques 6 - Views and Spreadsheets	76	Move	92
Views	76	Delete	92
Making a new view	76	Replicate	92
Closing a view	76	Moving references - an exact description	92
Discarding Spreadsheets	76		
Appearance of views	76	Techniques 12 - Use of Multiple spreadsheets	95
Scale view	77	Sheet naming convention	95
Grid	77	Using multiple sheets	95
Labels	77	An example of multiple spreadsheet working	95
Icon ribbon	77	The need for Recalculation	98
Formulae	77		
Techniques 7 - Creating and Destroying Big Cells	78	Techniques 13 - Saving, Loading and Workspaces	99
Creation of Automatic big cells	78	Saving	99
Creating and destroying Explicit big cells	78	Saving a sheet loaded then changed.	99
Multi-line text	79	File Information	100
		The Modification Marker	100
Techniques 8 - Number format dialogue	80	Hints on saving	100
Setting a Number format	80	Saving Selection	100
Type	81	Loading	100
Decimal Places	81	Workspaces	101
Before number	81	Saving a workspace	101
After number	81	Loading a Workspace	101
Negative display	82	More details about workspaces	101
Thousands separator	82	Resource files	102
Decimal Separator	82		
Alignment	82	Techniques 14 - Importing spreadsheet data	103
		File types for import	103
Techniques 9 - Custom formats	83	Importing Sprites and Drawfiles	104
Setting a Custom format	83	Importing TSV files	104
The meaning of a Custom format	83	Importing CSV files	104
Displaying a percent sign	83	Importing SID files	105
Alternative for zero	83	Importing WK1 files	105
Layout of numbers	83	Importing Schema Files	105
Layout of dates and times	83	Importing Schema Workspaces	106
Colour of Negative numbers	84	Using the clipboard to import data from other applications	106
Extra characters for negative values	85	Importing data in MS-DOS format	106
Thousands separator	85		
Fraction separator	85	Techniques 15 - Exporting spreadsheet data	107
Alignment	86	Exporting Drawfiles	107
Sequence	86	Exporting CSV files	107
		Exporting TSV files	107
Techniques 10 - Styles	87	Exporting WK1 files	107
Applying a style	88	Exporting Formulae	108
Creating a new style	88	Exporting Print text	108
Editing a style	89	Transferring files to the MS-DOS environment.	108
Deleting a style	89		
Removing Effects	89		
Setting a Labels style	89		
Sharing styles between sheets	89		
Techniques 11 - Editing tools	90		
The Clipboard	90		
Cut, Copy and Paste	90		
Cut	90		
Copy	91		
Paste	91		
Paste values -> Normal	91		
Paste values -> Transpose	91		

Techniques 16 - Printing	109	Techniques 21 - The Graph dialogues	126
Draft printing	109	Introduction	126
Image (normal) printing	109	Specifying a graph	126
The Print dialogues	110	Plotting a simple graph	126
Saving options without printing	110	New Graph dialogue	127
Printing a spreadsheet	110	Graph Data Dialogue	127
Draft	110	Graph Axes dialogue	128
Common Print options	111	Graph Options dialogue	128
Options for draft mode	111		
Options for the image mode	111	Techniques 22 - Basic Graph types	130
Selecting areas to print	112	Line plots	130
Controlling spreadsheet layout	113	Marked lines	131
		Scatter diagrams	131
Techniques 17 - Constants in Schema2	114	Vertical histograms	132
Numbers	114	Stacked Vertical histograms	134
Appearance of numbers	114	Horizontal histograms	134
Date and Time (Temporal) values	115	Stacked Horizontal histograms	135
Entering Dates	115	Pie Charts	136
Entering Times	116	Pictograms	137
Strings	116	General graphs	138
Block designations	116		
		Techniques 23 - Decorating Graphs	139
Techniques 18 - Operators	117	Selecting Objects	139
Arithmetic operators	117	Dragging objects	139
String operator	117	Graph Edit menu	139
Relational operators	118	Edit submenu	139
The NOT operator	118	The Effects Submenu	140
Bitwise logical operators	118	The Graph Submenu	141
Bitwise shift operators	119	Utilities menu	144
IF ... THEN ... ELSE ... ENDIF	119	Insert submenu	144
Using multiple operators and precedence	119		
Brackets	120	Techniques 24 - Storage of Graphs	146
		Discarding a graph	146
Techniques 19 - Sheet variables	120	File menu	146
Creating Sheet Variables	120	Sizing graphs	147
Editing Sheet Variables	121	Re-opening a hotlinked graph	147
Deleting Sheet Variables	121		
Using Sheet Variables	121	Techniques 25 - Evaluation	148
		Circular references	148
Techniques 20 - Standard Functions	122	Automatic and Manual Evaluation	149
Arguments and results	122	Intelligent and Mechanical evaluation	149
The Function Groups	123	Evaluation by row and by column	149
Time functions	123	Evaluation mode dialogue	150
String functions	123	Manual evaluation	150
Scientific Functions	124	Recalculate all	150
Trigonometric functions	124	Values	151
Array functions	124	Final comment	151
Random number generators	125		
Algebraic functions	125	Techniques 26 - Default documents and Choices	153
Financial functions	125	Default document	153
Statistical functions	125	Changing the default document	153
Matrix functions	126	Choices dialogue	154
Standard macros	126		

Techniques 27 - Resource files	155	Appendix A	161
Loading resource files with a workspace	155	Definition of Schema2 Standard Functions	161
Menu Description Files	155	Part 1 - Overview	161
Labels with Parameters	157	1. Introduction	161
Editing a Menu Description file	158	Part 2 - Algebraic Functions	162
Deleting Items	158	Part 3 - Scientific Functions	163
Changing keyboard short cuts	159	Part 4 - Trigonometrical Functions	164
The Icon Ribbon	159	Part 5 - String Functions	166
Altering the icon ribbon	160	Part 6 - Array functions	168
To add a new icon	160	Part 6 - Array functions	168
The Pictures directory	160	Part 7 - Time functions	170
Modifying the resource files in !Schema2	160	Part 8 - Random functions	172
		Part 9 - Statistical functions	173
		Part 10 - Financial functions	175
		Part 11 - Matrix functions	179
		Part 12 - Miscellaneous functions	181
		Part 13 - Lookup Functions	182
		Part 14 - Database operations	184
		Part 15 - Useful Macros	188
		Group 1:Financial macros	188
		Group 2: String macros	188
		Group 3 : Miscellaneous functions	189
		Part 16 - Lotus 1-2-3 macros	190
		Part 17 - Alphabetic list of functions and macros	195
		Appendix B - Definitions of TSV and CSV files	198
		TSV Files	198
		CSV Files	198
		Appendix C - Schema2 and Lotus 1-2-3	200
		Introduction	200
		Recommendations for Lotus1-2-3 users:	200
		Recommendations for Schema2 users	200
		Appendix D - Notes for users transferring from Schema	202
		Filetypes	202
		Selection	202
		Entering data and formulae	202
		Copy and Move	202
		Styles	202
		Graphs	202
		User names	202
		Appendix E - Keyboard short cuts in Schema2	203

Preface

This tutorial gives a gentle introduction to most of the important concepts and techniques of Schema2. The text is divided into two main parts:

Part 1 is a sequence of ten linked examples which show you how to enter and display data, arrange calculations, print, draw graphs, and use many other features of spreadsheets. When you complete Part 1 you will have a good basic knowledge of Schema2, and you'll be able to handle most everyday spreadsheet applications.

If you already know how to use a spreadsheet, you will find Part 1 of the manual easy. It is still worth reading, because it describes many new features which help you get the best out of Schema2.

Part 2 of the tutorial contains straightforward descriptions of more advanced topics. Be selective in what you read; use the index at the beginning to pick out the sections you need.

To use Schema2 you must be familiar with the **RISC OS** operating system. You should know how to create directories and run applications, and you must feel comfortable using windows, a mouse, icons and menus. If you don't know about these techniques, look in your user's guide, **Welcome to RISC OS**.

If you use **Schema** (which is a totally different product from Schema2) read the notes in Appendix D.

As you read the material, run the examples on your computer - this is a great help in making everything make sense. You will need to use the tutorial files which can be found in the **Tutorial** directory on Resource Disc 1. Make sure that you have this directory (or a copy) to hand when using the Tutorial.

Introduction to Schema2 Macros

This booklet, which was previously supplied as a separate volume and had to be applied for separately, is now included and will be found after the end of the main Schema2 manual.

Typographic conventions

- Mouse buttons appear like this **SELECT**
- Instructions to press a key on the keyboard are shown like this, eg. **Return** means press the Return key on the keyboard.
- Menu items are shown in bold, 'Click on **Info**' means choose the item called Info from a menu.
- Menu items which lead to a sub-menu have an arrow after them, like this **->**, so 'Click on **File -> Save**' means 'Open the sub-menu from the item **File** and click on **Save** on the sub-menu'.
- Something important always appears **in bold**.

Introduction to Spreadsheets

Few people manage to escape the tyranny of calculation. At some time in our lives most of us are obliged to take up a pencil and paper and do sums. You might be checking the total on a restaurant bill, or calculating the amount of nutrients and fibre in your daily diet, or you might want to work out how much wine you can buy on your holiday and still leave you with enough money to get home.

Nowadays, pocket calculators have taken the drudgery out of ordinary arithmetic, but you still need pencil and paper to write down the answers and organise the calculation. Furthermore, hardly any electronic calculators let you check that you have entered the numbers correctly; to be sure of your answers you really need to do everything twice.

The spreadsheet program lets you use a computer for sums just as simply as a calculator. There are some other advantages: you get a permanent record of the numbers you put in, as well as the method used to work out the results.

The spreadsheet is a computerised version of the 'calculating sheet' traditionally used by accountants.

The old-style calculating sheet is a large oblong sheet of paper. Vertical and horizontal lines divide the sheet into a grid of 'cells'.

To use the sheet, the accountant pencils in some row and column headings, and then fills in columns of figures, such as financial records. He then adds up each column, and may carry forward the result to the next column. The figure below shows an example of this kind of sheet.

	A	B	C	D	E	F	G	H
1	Sales Figures for 1993							
2								
3		Jan	Feb	Mar	Apr	May	Jun	Jul
4								
5	Glasgow	4562.95	3317.48	151.64	4412.07	4265.71	2370.66	3406.25
6	Northwich	3375.79	4388.01	608.38	1562.74	2520.97	3814.78	1145.07
7	Cambridge	564.42	4868.55	1673.48	276.15	3297.21	4073.68	28.96
8	Totals	8503.17	12574.04	2433.50	6250.97	10083.89	10259.12	4580.27

Over the years, calculating sheets became widely used for problems which involved recording numbers and doing calculations. Examples are exam and class records, sports results, business financial forecasts, domestic finances, expedition planning, and weather records.

The vital difference between a paper calculating sheet and a spreadsheet lies in the way that cells are used to record *calculated* figures. With a paper sheet, you have to work out the numbers yourself. With a computer-based spreadsheet, on the other hand, you simply type in a formula and the machine works out the answer and fills in the cell for you. Furthermore, if you change one of the numbers used in the formula, the computer immediately repeats the calculation and updates the answer automatically.

The Examples

All ten examples in this introduction are based round the world-famous College of Irrational Sciences at Forres in the Scottish Highlands. Three of the staff there are:

- Ms. Rona Toner, a lecturer in charge of several courses.
- Mr. Hamish MacRihanish, in charge of Student Liaison.
- Dr. Quentin Curser, the Bursar.

We will see how the College uses Schema2 to help run its affairs.

When you read the examples, you should work them through on your computer.

Some actions are printed in italics; these are the ones you must carry out to gain the full benefit of the example.

The College of Irrational Sciences is entirely fictitious. None of its staff or students are real characters, and any similarity with living people is coincidental. The reports of their activities are not intended to be believed - but they *do* know how to use Schema2.

Examples

Example 1 - Class Records

This example discusses:

- opening a new spreadsheet
- moving about the sheet
- entering and correcting data
- saving the sheet as a file
- discarding the sheet
- ending a Schema2 session
- reloading an existing spreadsheet
- selecting a block
- printing the spreadsheet
- using keyboard short cuts to speed up menu selections

In this example you'll see how Rona Toner can use a spreadsheet for one of the simplest applications at the College of Irrational Sciences - one that does not need any calculation at all. Rona takes advantage of the regular structure of the spreadsheet to record and print information as a table.

Rona is in charge of a Water Divining course. There is a practical class every week, but some of the students are lazy and don't always bother to come. When Rona joined the college staff a few months ago, the Principal told her to be sure to keep track of student attendance. Rona uses a spreadsheet for weekly records, which show whether each of her students turns up on time, comes late, gives a reason for not coming or simply fails to appear. Rona started up her spreadsheet by following these instructions. Do the same:

Ex 1.1 Starting up and opening a new sheet

Double click on the !Schema2 file icon and wait until the program appears on the icon bar. Then click on the program icon. A sheet like that shown in Figure 1.1 will appear.

The sheet appears in an ordinary RISC OS window. You can move the window around the screen using the title bar, or change its size as usual, close it with the 'close' icon, or send it to the back of the screen with the 'back' icon.

The title bar shows the name of the sheet, which is currently **Untitled** and the magnification, which is presently 100% or 'full-size'.

The window is divided into five areas or *panes*.

Just under the title bar is the **input area**. This has three parts.

At the top is a white band where you enter numbers, text or formulae.

Below this there are five special icons and a box which holds a cell name such as **A1**.

Below the five special icons is another band of icons, which can be used to start common actions like printing.

Next, working down, comes the **top margin**. This holds boxes with the letters **A** to **F**.

On the left of the sheet, below the input area, is a small rectangle called the **neutral area**. Beneath comes the **side margin**, which holds the numbers **1** to **6**.

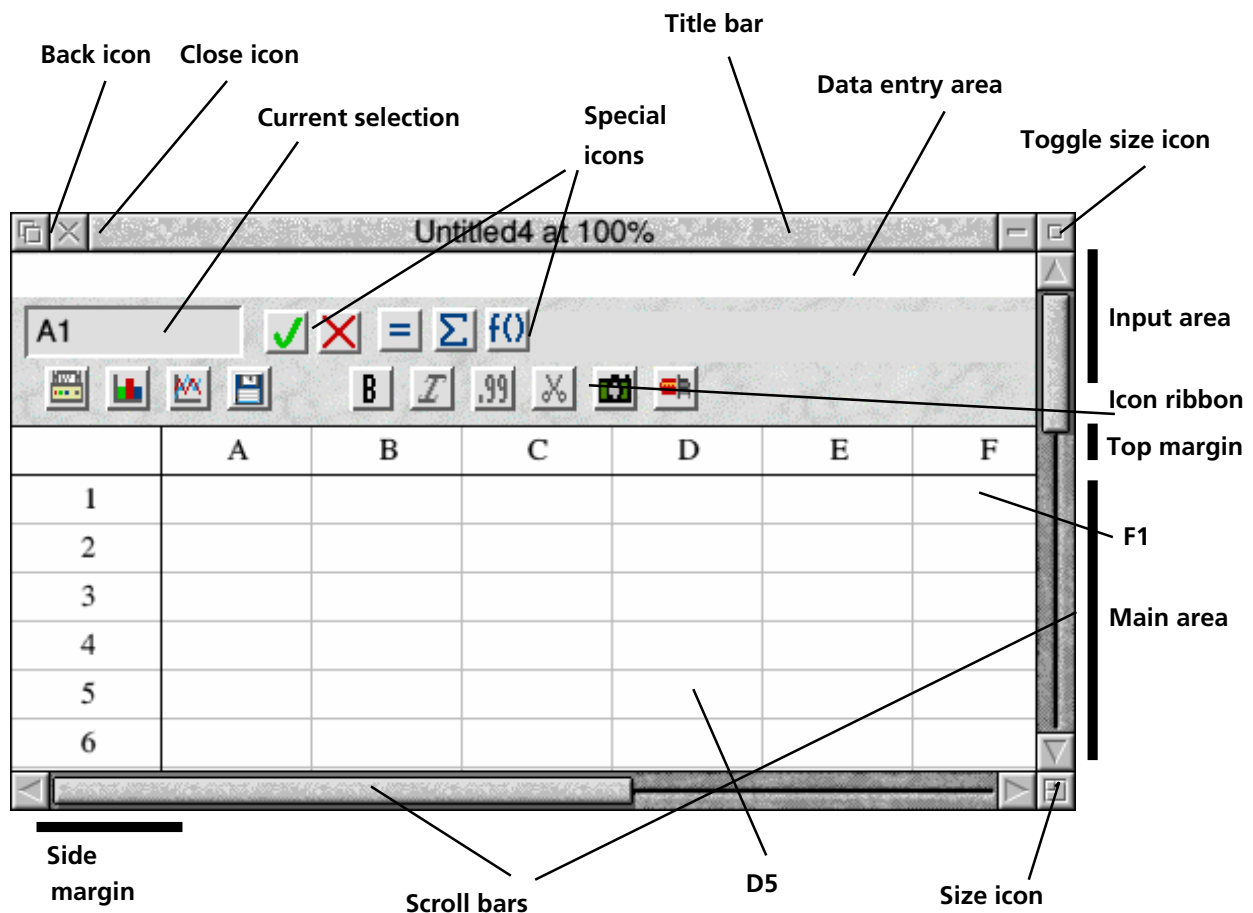


Figure 1.1

Apart from these small panels, the bulk of the sheet, called the **main area**, is filled with blank squares. Each square or cell has a name given by its column and row. Figure 1.1 shows you where to look for **D5** and **F1**.

The cell with the dark background is called the **selected cell**. Its name appears in the box in the input area.

In Schema2, spreadsheets are not limited by the window size but can be big enough for all practical applications. Even the sheet you have just opened is rather bigger than the window it appears in. You can reach other parts of the sheet by scrolling. However, when you scroll, it is worth noticing that:

- the input area never moves: it remains at the top of the spreadsheet window.
- the top margin does not scroll off the top of the window but only moves sideways.
- the side margin does not scroll off the side of the window but only moves up and down.
- only the main area moves in both directions.

This arrangement ensures that no matter how far you scroll a sheet, you can always see both the input area, and the column and row names for the cells currently in the main area.

Try scrolling the sheet so as to reach cell **J22**. As you will see, only the main area is allowed to move in both directions. This is shown in Figure 1.2.

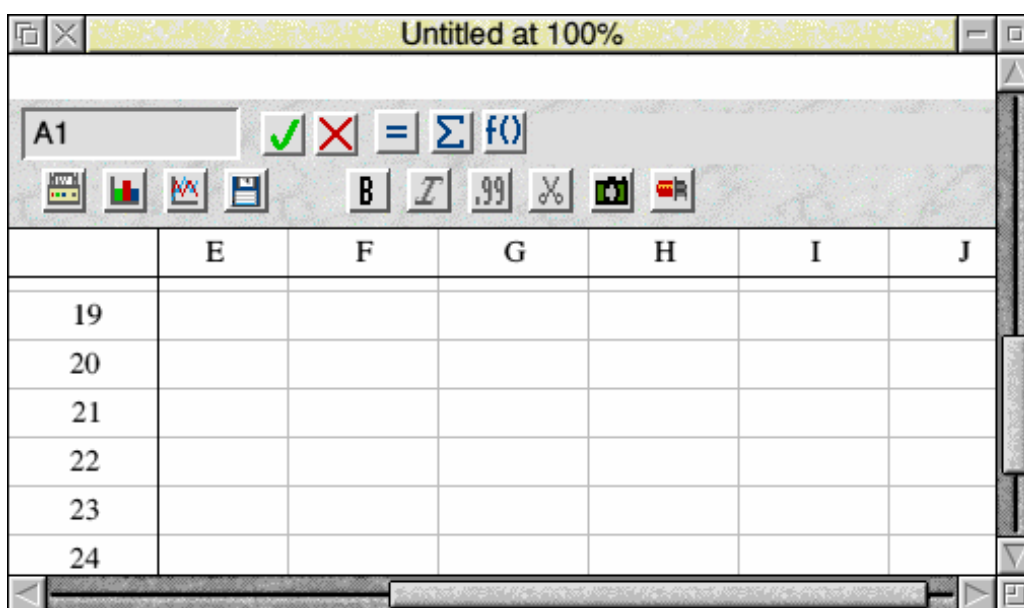


Figure 1.2

Before Rona filled in the Water Divining Class Record spreadsheet she wisely planned its layout. She decided that she would:

- keep the top row (row 1) for a heading.
- use the next row (row 2) for dates.
- use column A for the students' first names, starting at row 3.
- use column B for the students' surnames, lining them up with the first names in column A.
- Use the rest of the columns from C onwards (one column per week) to record attendances.

Rona's students' names are:

Debra De'Ath	Myra Palmyra
Krystle Ball	Rose Lee
Jocasta Spell	Marie Provan
Arthur Crown	Rava Strange
Odin Rune	

Rona defined this simple code to show attendance:

P: Present
 L: Late
 E: Absent, with excuse
 X: Absent, no reason given

Ex 1.2 Entering text

This is how Rona started her sheet. Do the same using Figure 1.3 as a guide.

1. Click **SELECT** on cell C1. If you have scrolled the sheet you may have to move it back to bring the cell into view. The cell will go black, showing that it is 'selected' and its name (C1) appears in the box in the input area.
2. With cell C1 selected, type the heading

Water Diving Records

When you type, characters appear in the input area, and you can use the **Delete** key to correct any mistakes. The cursor keys move you back and forth along the input line.

3. When the heading is correctly entered in the input area, press **Return**. The heading appears in cell C1, which expands so that there is room for all the characters. The cell below (C2) turns black, showing that it has been selected for the next entry.
4. Type **June 12** and then press **Tab**. This enters the information into cell C2 and selects D2.
5. Type **June 19** and then press **Tab** to enter the date and select E2.
6. Type **June 26** and then press **Tab** to enter this date and move to F2.
7. Now finish off filling in the rest of the spreadsheet using the information in Figure 1.3. You'll find it easiest to type all the first names into column A, and then the surnames into column B.

	A	B	C	D	E
1			Water Diving Records		
2			June 12th	June 19th	
3	Debra	De'Ath	P		
4	Myra	Palmyra	P		
5	Krystle	Ball	X		
6	Rose	Lee	E		
7	Jocasta	Spell	P		
8	Mire	Provan	L		
9	Arthur	Crown	P		
10	Rava	Strange	P		
11	Odin	Rune	E		
12					

Figure 1.3

Rona has noticed that there are three ways of entering the information using the keyboard.

- If she clicks **SELECT** on the **tick icon** in the special icon bar (see Figure 1.3) the information is entered and the cell stays selected
- If she presses **Return** the information is entered into the selected cell and the cell below is automatically

selected.

- If she presses **Tab** the information is entered into the selected cell and the cell to the right is selected.

Ex 1.3 Correcting mistakes in spreadsheet cells

You may have spotted a deliberate mistake in our spreadsheet - the name **Mire** is actually meant to be **Marie**. You can correct this error as follows:

1. Click **SELECT** on the bad cell (**A8**) to highlight it. The contents of the cell - **Mire** - appear in the input area.
2. Put the mouse pointer between **r** and **e** in the input area, and click **SELECT**. Type **i** to insert the new character, giving you

Mirie

3. Use the left cursor control key to move the cursor back two spaces so that it comes between the first **i** and the **r**.
4. Press **Delete** to remove the **i**, and then type **a** to insert an **a** in its place. This corrects the entry, giving **Marie**.
5. Press **Return** to put the new entry back in the selected cell.

As an alternative, you can erase the whole entry (in two ways) and type it afresh:

1. Type **Ctrl-U** to clear the text area, or Click **Select** on the **cross icon** to clear the text area.
2. Type the word **Marie** and press **Return** or **Tab**, or click on the **tick icon**.

You must always remember to click on the **tick icon**, or to press **Tab** or **Return** after keying in information. If you forget, the information in the input area is not entered into the spreadsheet cell.

If you try to select another cell with the mouse before you enter existing information, the cell will not be selected - instead, its name appears in the text area. For example, you might get something like

Jim Bloggs +B7+B7+B7+B7

as you click frantically on **B7**. This is a common error, and the best way out is to rub out the "+B7"s using the **Delete** key, and then press **Return** or **Tab**.

Ex 1.4 Filing or Saving a spreadsheet

When Rona filled in the first week's attendance records, she filed the sheet so that she could bring it back the following week and update it. She decided to call the spreadsheet **Water**.

1. Click **Menu** over the body of the sheet. The spreadsheet menu appears.
2. Use the mouse to move through the menu, and choose **File -> Save**. A dialogue box appears with the suggested filename: **Untitled** (Figure 1.4).

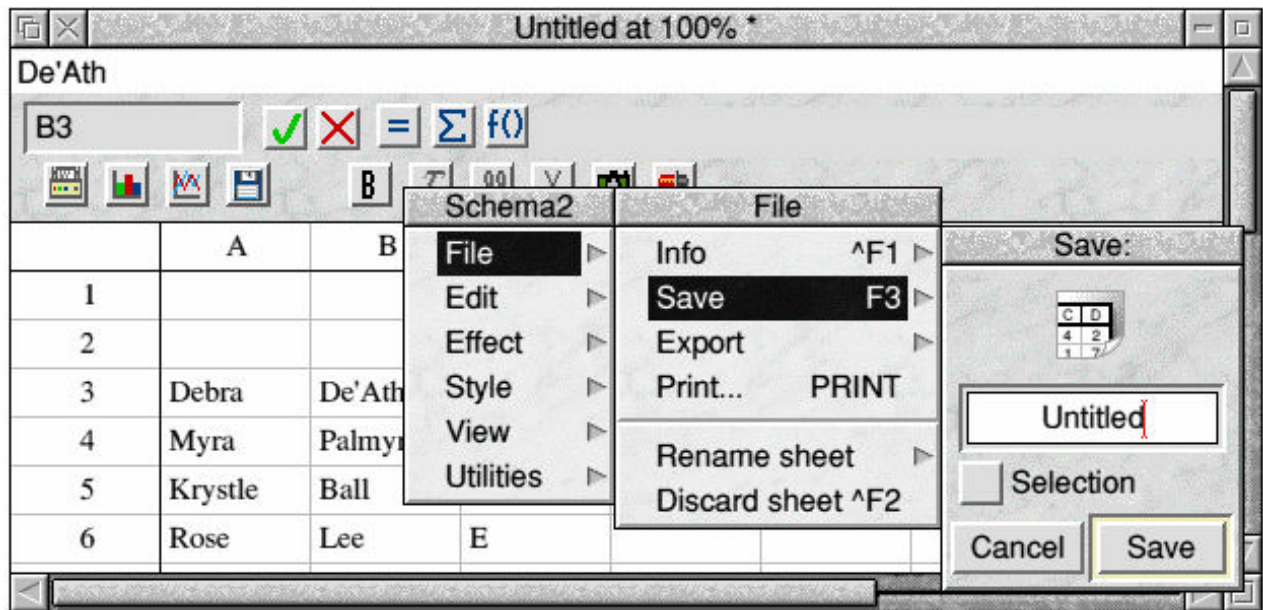


Figure 1.4

3. Press **Ctrl-U** to clear this name and type **Water** to name the spreadsheet.
4. Drag the file icon from the box to the directory window which you set up when you started. If you don't have the **Practical** directory window ready, don't worry - just cancel the Save operation by clicking anywhere off the dialogue box, set up the directory, and start the process again.

When Rona saved her spreadsheet its new name appeared on the title bar.

Ex 1.5 Closing a spreadsheet

Once Rona saved the sheet, she could close it down and end her Schema2 session. Here is what she did:

1. Bring up the Schema2 icon bar menu, move to the **Discard** submenu and select **Water**. Alternatively you can open the main sheet menu and choose **File -> Discard sheet**.

If Rona ever tries to discard a sheet which has been created or altered but not saved, **Schema2** will show a dialogue like this.

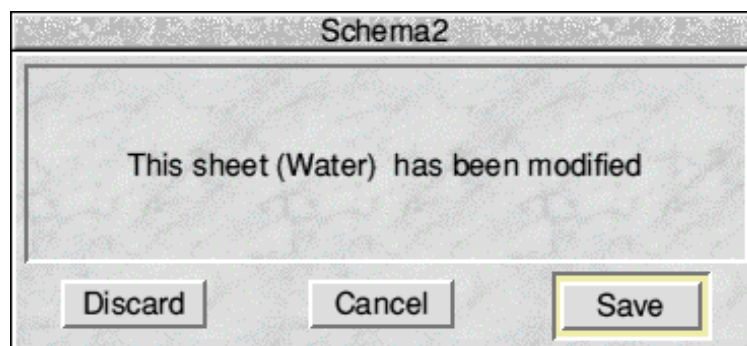


Figure 1.5

Rona must choose one of three possible courses:

- She can go ahead and Discard the sheet. This will lose any changes she has made.
- She can Cancel the whole operation.
- She can Save the sheet.

Rona selects the right course by clicking on one of the buttons on the dialogue.

Ex 1.6 Ending a Schema2 session

Rona's instructions for ending a Schema2 session are given below.

1. Click **MENU** on the Schema2 icon, to display the icon bar menu.
2. Choose Quit from the icon bar menu.

If Rona tries to quit from Schema2 when there are still some unsaved sheets around, she will get a warning dialogue like this. (Figure 1.6)

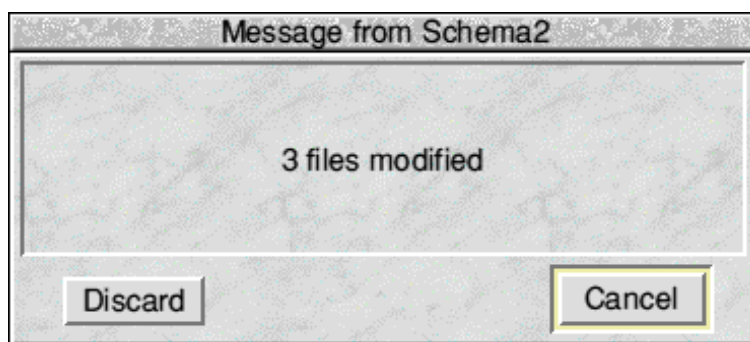


Figure 1.6

Rona can now decide whether to discard all the sheets or cancel the Quit operation and save the files one by one.

Ex 1.7 Reloading a spreadsheet

A week has passed, and Rona has some more records to enter. There are two ways she could re-load the sheet she filed the week before:

Method 1

This method works whenever Schema2 is already loaded, so that Rona can have many sheets open at the same time.

1. Drag the **Water** icon from the **Practical** directory to the Schema2 icon on the icon bar.

Method 2

1. Double click **SELECT** on the **Water** icon.

For this method to work RISC OS must have 'seen' Schema2 in this session; that is, the Schema2 icon must have appeared in a directory window. The double click loads Schema2 (if it is not already loaded) and brings up the **Water** spreadsheet.

Now Rona can enter the second week's records, which are:

Student	Attendance	
Debra De'Ath		P
Myra Palmyra		P
Krystle Ball		P
Rose Lee	P	
Jocasta Spell		P
Marie Provan		P
Arthur Crown		L
Rava Strange		X
Odin Rune		P

When Rona entered the second week's records into the Water spreadsheet she needed to save the new version. If you look at the title bar you will see that there is now an asterisk after the title **Water**. This shows that the spreadsheet has been newly amended, and needs to be saved again before it is closed.

After three weeks, Rona's spreadsheet looks like this (Figure 1.7):

	A	B	C	D	E
1			Water Divining Records		
2			June 12th	June 19th	June 26th
3	Debra	De'Ath	P	P	P
4	Myra	Palmyra	P	P	L
5	Krystle	Ball	X	P	X
6	Rose	Lee	E	P	X
7	Jocasta	Spell	P	P	P
8	Marie	Provan	L	P	P
9	Arthur	Crown	P	L	L
10	Rava	Strange	P	X	E
11	Odin	Rune	E	P	P

Figure 1.7

Ex 1.8 Saving the latest version of your spreadsheet

Rona could save the latest version of her spreadsheet in three ways.

Method 1

1. Click **MENU** over the body of the spreadsheet to call up the menu.
2. Navigate through the menu and choose File -> Save. This brings up the **Save** dialogue.
3. Click **SELECT** on the **OK** box in the Save dialogue.

There is no need to drag the icon to a directory window: this is only necessary when you are saving a new sheet, or a sheet you want to rename.

Method 2

1. Press the **F3** function key. This brings up the Save dialogue box.
2. Press **Return**. This has the same function as clicking the **OK** box in the Save dialogue box.

Method 3

1. Bring up the main menu and click on File -> Save. This saves the sheet without showing you the dialogue.

Ex 1.9 Selecting a block of cells

As Rona uses Schema2, she often needs to select a **block** or group of cells. Blocks are always rectangular. When a block is selected, all the cells in it turn black so that she can easily see which cells she has chosen.

We shall be showing you various ways to select blocks. Perhaps the simplest method is to click **SELECT** on the cell in one corner, and then click **ADJUST** on the cell in the opposite corner. For example, suppose Rona clicks **SELECT** on cell **B3** and then clicks **ADJUST** on cell **F6**. This selects the block of 20 cells with corners at **B3**, **B6**, **F6** and **F3**. Try it!

Ex 1.10 Printing Spreadsheets

After three weeks Rona wanted to print her records for public display. At this point the sheet was all blank except for the columns **A** to **E** and rows **1** to **11**; this is the only part of the sheet which needed to be displayed.

Here is how she printed the interesting part of the sheet:

1. Make sure that a printer driver or manager is loaded.
2. Select the block to be printed. Click **SELECT** on cell **A1** and then click **ADJUST** on cell **E11**. Rows **1** to **11** in columns **A** to **E** are now selected. All the cells in the selected block turn black.
3. Open the main menu and choose **File -> Print**. You will get a dialogue box like this (Figure 1.8):
4. Click **SELECT** on the Selection button and then on the **Print** button.

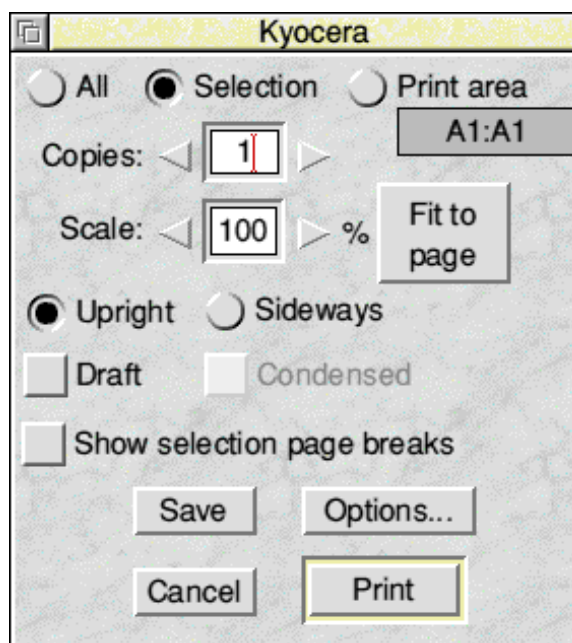


Figure 1.8

Ex 1.11 Using keyboard short cuts

In Example 1.9 we mentioned that you can use the **F3** function key to bring up the Save dialogue. **F3** is often called a *hot key*, because it replaces a menu selection and you can use it without taking your hand away from the keyboard. If you know which key to use it is faster to press a key than to steer your way through a set of menus and choose an item. You will find that most of the menu entries in Schema2 have keyboard short cuts assigned to them. The keys are all non-printing characters, so there is never any confusion between data and commands.

Keyboard short cuts fall into several groups:

Keys used by themselves: This includes the function keys **F1** to **F11** and the **Print** key.

Keys used together with the Shift key: They include **F1** to **F11**.

Keys used together with Shift and Ctrl: **F1** to **F11**.

Keys used with Ctrl: This includes **F1** to **F11** and most of the letters.

When a menu item has a keyboard alternative, the name of the key is displayed in the menu. Some examples are:

Save	F3	(F3 means Save)
Info	^F1	(Ctrl-F1 calls for information)
Input Block	↑^F8	(Shift-Ctrl-F8 calls for an input block)
Bold	^B	(Ctrl-B specifies bold characters)

Wherever possible keyboard short cuts follow Acorn standards; for example, **Ctrl-U** clears the current line, and **Ctrl-C** is a copy command.

At this stage, you may not understand what these commands actually do. This doesn't matter - all you need to know is that many commands have keyboard short cuts, and that **^** stands for the **Ctrl** key and **↑** for the **Shift** key. As you learn the various Schema2 commands, you'll pick up the keyboard short cuts almost without thinking.

Summary

You can now create a new spreadsheet, fill its cells with text, print it, file it away and retrieve it to make more entries. You understand how to correct typing errors and how to navigate around the sheet, choosing any cell you need, and you know how to use keyboard short cuts.

Example 2 - How long is a piece of string?

This example covers:

- entry of arithmetical formulae
- automatic evaluation

When students graduate from the College of Irrational Sciences, each one is sent a parcel with the tools of their future trade - it could be a crystal ball, a pack of Tarot cards, a divining rod, a box of special-purpose tea, a spelling book, or any combination of these items depending on what courses the student has taken.

The parcels are wrapped and tied up in the Bursar's department. Dr. Curser hates waste and insists that the string used for each parcel should be cut to exactly the right length. He uses a spreadsheet to decide just how much string is needed for a parcel of any shape and size.

Dr. Curser's staff always tie parcels in the way shown in Figure 2.1.



Figure 2.1

The string runs up all four sides of the parcel, and across the top and bottom, in both directions. The knot takes 7 inches of string.

Adding up all these lengths, it turns out that the string must be

	four times the height of the parcel
plus	twice the length
plus	twice the width
plus	7 inches (for the knot)

This translates into the formula

$$\text{Length of string} = 4 \times \text{height} + 2 \times \text{length} + 2 \times \text{width} + 7$$

This formula is a rule for working out the length; whatever the size of the parcel it will always give you the exact length of string needed.

In Schema2, formulae are almost the same as the ones you find in ordinary algebra. However, there are three main differences:

- All formulae start with the = sign. This signals to the computer that what comes after the equal sign has to be calculated, rather than simply entered into the sheet.
- Algebraic names like height, length or width are not used. Instead the names of the cells which hold these quantities are written: this might be **A2** for the height, **B2** for the length and **C2** for the width.
- In Schema2, you must show multiplication with the * sign (not x).

Using these rules, the spreadsheet formula to work out the length of the piece of string looks like this:

$$=4*A2 + 2*B2 + 2*C2 +7$$

Ex 2.1 Setting up a sheet with a formula

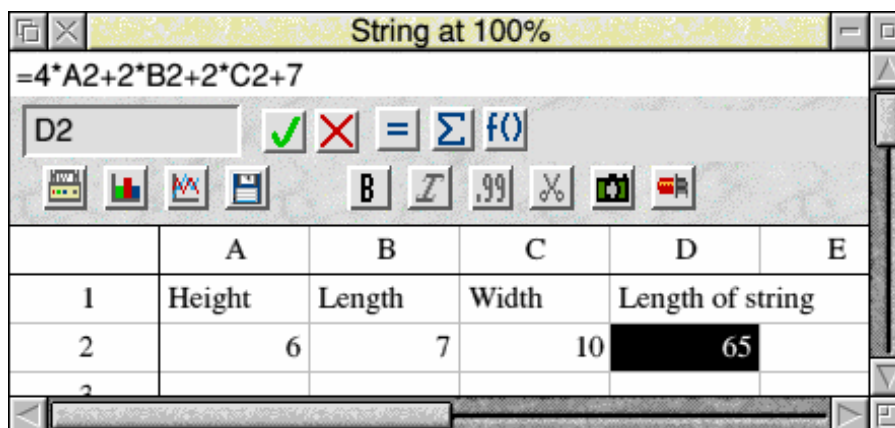


Figure 2.2

Dr. Curser's string spreadsheet is shown in Figure 2.2. Here is how he built it for a box 6 inches high, 7 inches long and 10 inches wide.

1. Open a new spreadsheet.
2. Set out the spreadsheet as shown in Figure 2.2. Put the column headings in row 1 and the dimensions of the box in row 2, using cells **A2** to **C2**.
3. Click **Select** on cell **D2**. Type the spreadsheet formula

$$=4*A2 + 2*B2 + 2*C2 + 7$$
and press **SELECT** on the tick icon.

If you have done all this correctly, the answer (65) appears in cell **D2**! Schema2 has used the data values in cells **A2**, **B2** and **C2**, substituted them into the formula you placed in **D2**, and displayed the result in **D2**.

Ex 2.2 Checking Formulae

As it stands, the spreadsheet does not display the formula explicitly. To check it is right, Dr. Curser clicks on **D2**, and the formula reappears in the input area. If it is wrong he can change it and the new result is displayed immediately.

Ex 2.3 New Data

Dr. Curser usually sends parcels in batches, all of different shapes and sizes. He will generally need a different length of string for each parcel.

To work out the length for any parcel, he only needs to change the numbers in cells **A2** to **C2**. The new answer comes up in **D2** automatically. This feature is called **Automatic Evaluation**.

Try this calculation:

*To find the length of string you need for a parcel 20 inches high, 23 inches long and 30 inches deep, enter 20 into cell **A2**, 23 into cell **B2** and 30 into cell **C2**. Then read the answer from cell **D2**.*

When Dr. Curser has cut all the strings he needs for a batch of parcels, he saves the spreadsheet under the name **String**. Then he can use the sheet again next time he has any parcels to send.

Summary

This simple example has demonstrated some of a spreadsheet's most useful features:

- Each cell can hold text or a number. It can also hold a formula, which is used to calculate numbers.
- When you change any cell, all the formulae which depend on that cell are recalculated automatically and their results displayed straight away. This is called intelligent evaluation.

In this example we have looked at the entry of arithmetical formulae and automatic evaluation.

If you have followed the explanation so far, you have already learned most of the key points about spreadsheets. You could now set up a spreadsheet of your own to organise a school timetable or predict your company's future. However, we recommend that you continue with the rest of this tutorial, which introduces you to some useful features which help to set up a spreadsheet, improve accuracy, give the layout a better appearance, and plot graphs.

Example 3 - Exam results

This example covers:

- block entry of formulae
- locking rows and columns
 - unlocking rows and columns

In this example Hamish MacRihanish sets up a typical spreadsheet to do simple arithmetic on lots of numbers - in this case, examination marks.

At the College of Irrational Sciences, students take four main subjects :

Astrology
Palmistry
UFOlogy
Sorcery

At the end of the academic year students take exams in all four. To obtain the Diploma of Irrational Science each student must get an average of 50%.

Hamish MacRihanish, the Student Liaison Officer, has the task of working out overall results and giving students the news. He finds the spreadsheet a great help. Once it is set up he can just enter the marks for each examination, print the sheet and put it up on the notice board. Figure 3.1 shows part of a spreadsheet for the exam results.

1			College of Irrational Sciences' Exam Results			
2			Astrology	Palmistry	UFOlogy	Sorcery
3	Debra	De'Ath	25	49	64	40
4	Myra	Palmyra	89	91	62	78
5	Krystle	Ball	70	35	55	62
6	Rose	Lee	71	11	60	49
7	Petra	Black	68	71	38	51
8	Marie	Provan	52	42	68	91
9	Robert	Falconer	37	31	34	50
10	Andreas	Colin	7	12	8	47
11	Michael	Lamont	42	119	60	49
12	Veronica	Rowout	68	84	95	89
13	Alison	Haywain	51	61	56	56
14	Michael	Rooney	52	78	36	41
15	Fred	Grant	49	82	28	16
16	Solomon	Rooney	73	66	59	81
17	Zuleika	Ball	46	65	51	48
18	Geoffrey	Quincy	51	17	40	58
19	Ebenezer	Rooney	70	55	42	39
20	Lucy	Borgia	76	40	36	71

Figure 3.1

- Row 1 is used for a title.
- Row 2 contains the names of the subjects and the column heading for the average marks.
 - Columns A and B are used for the students' first and second names.

Hamish has already filled in the students' names, the subject names and the exam marks. All that is needed is to

put in a formula to work out the average mark for each student, and the overall result.

*Load the spreadsheet, which you will find in sub-directory **Tutorial** with the name **ExamMarks**, and look at it in more detail. As you can see there are 18 students in the year. There are two sisters from the same family and three brothers from another family. The spreadsheet is not set out in alphabetical order but in the random order in which students registered for the course. We suggest you scroll down the spreadsheet to see who was last to join.*

Hamish plans to display the average marks in Column **G**, starting in cell **G3** and running down to **G20**. To find the average of the four subject marks for each student, he needs to add the marks together and divide by four:

$$\text{Average} = \frac{\text{Astrology} + \text{Palmistry} + \text{UFOlogy} + \text{Sorcery}}{4}$$

4

Column **G** must be filled with suitable formulae, one for each line:

G3: = (C3+D3+E3+F3)/4 for Debra De'Ath
G4: = (C4+D4+E4+F4)/4 for Myra Palmyra
G5: = (C5+D5+E5+F5)/4 for Krystle Ball
G6: = (C6+D6+E6+F6)/4 for Rose Lee

and so on.

Note the use of the brackets in the formulae. They ensure that Schema2 adds up the scores before dividing the total by 4. In general, Schema2 follows convention and evaluates parts of a formula inside brackets first.

Don't type in all these formulae now: We are about to demonstrate a quicker and easier way to enter them.

Ex 3.1 Block Entries

As there are many people in the class, typing all these formulae one by one would be unutterably boring. Hamish also realises that he might make typing mistakes which could easily lead to a student being given the wrong result. To avoid all this repetition, Schema2 has a 'block entry' feature that lets you put copies of the same formula into all the cells in a **block**.

A block can be any rectangular collection of cells, such as all or part of a row, all or part of a column, or parts of adjacent columns.

When block entry is used the formulae entered into the cells are not always exactly the same; they are *adjusted* so that each cell refers to other cells in the same *relative* position. The need for adjustment is clear from the formulae earlier in the example:

The formula in **G3** refers to cells **C3**, **D3**, **E3** and **F3** which are the four cells to its left, the formula in **G4** refers to cells **C4**, **D4**, **E4** and **F4**, the four cells on *its* left, and so on all the way down the column.

To make a block entry here is what Hamish does.

1. Select block **G3..G20**. One way to do this (just in case you've forgotten) is to select cell **G3** and then click **Adjust** on **G20**.
2. Check that the box in the input area says: **G3..G20**
3. Now type: **=(C3+D3+E3+F3)/4**

This is a formula suitable for the first cell (**G3**) in the selected block. Remember, a formula must be preceded by an equals sign.

4. Press **Return** or click on the tick icon. The **Modify Block** dialogue box is shown to warn you in case you have blocked a group of cells by mistake.
5. Click on the Accept button.

A copy of the formula is entered into every cell in the selected block. The copy is adjusted according to its position in the spreadsheet. In each case, the formula in any cell refers to the four cells on its left in the same row.

The average mark for each student is calculated immediately, and the sheet now looks like Figure 3.2 - you can see straight away which student passed, and which one failed.

	A	B	C	D	E	F	G
1			College of Irrational Sciences' Exam Results				
2			Astrology	Palmistry	UFOlogy	Sorcery	Average
3	Debra	De'Ath	25	49	64	40	44.5
4	Myra	Palmyra	89	91	62	78	80
5	Krystle	Ball	70	35	55	62	55.5
6	Rose	Lee	71	11	60	49	47.75
7	Petra	Black	68	71	38	51	57
8	Marie	Provan	52	42	68	91	63.25
9	Robert	Falconer	37	31	34	50	38
10	Andreas	Colin	7	12	8	47	18.5
11	Michael	Lamont	42	119	60	49	67.5
12	Veronica	Rowout	68	84	95	89	84
13	Alison	Haywain	51	61	56	56	56
14	Michael	Rooney	52	78	36	41	51.75
15	Fred	Grant	49	82	28	16	43.75
16	Solomon	Rooney	73	66	59	81	69.75
17	Zuleika	Ball	46	65	51	48	52.5
18	Geoffrey	Quincy	51	17	40	58	41.5
19	Ebenezer	Rooney	70	55	42	39	51.5
20	Lucy	Borgia	76	40	36	71	55.75

Figure 3.2

Ex 3.3 Locking rows and columns

When you design a spreadsheet it is usual to use one or two columns at the left as row labels. For example, each row in the examination schedule is labelled with the student's name. Similarly, the top row (or rows) is used to hold column labels.

This arrangement starts to give problems as soon as the spreadsheet outgrows its window. As you scroll about the spreadsheet to bring up particular cells, the rows and columns you have used as labels may disappear altogether. For example, when you enter marks for the 20th student in the class you will not have any clear indication of what columns **C** or **E** actually stand for.

Schema2 allows you to 'lock' certain groups of rows or columns, so that copies of them appear in the margin and are always on view.

In the exams sheet, Hamish decides to lock row **2** and columns **A** and **B**, so that he can always see what subject and what student each figure relates to. Here is what he does:

1. Select the whole row of row **2** by clicking on its margin (the number '2').
2. Click **MENU** on the spreadsheet and choose Edit -> Rows -> Lock Rows.

A copy of the selected row appears in the top margin. When Hamish scrolls the spreadsheet up and down, this copy stays at the top of the display so that each column always has the right heading.

To lock the columns with the student names:

1. Select the whole of columns **A** and **B**. An easy way to do this is to click **SELECT** on the column **A** margin (letter A) and then click **ADJUST** on the column **B** margin (letter B).
2. Click **MENU** on the spreadsheet and choose **Edit -> Columns -> Lock Columns**.

In general, you can use the same method to lock any group of consecutive rows or columns. The sheet with locked rows and columns is shown in Figure 3.3.

Hamish soon discovered that the locked rows and columns were useful for entering data, but didn't improve the appearance of the printed sheet (compare Figures 3.2 and 3.3). To unlock the rows and columns, this is what he did:

1. Open the main menu and choose **Edit -> Rows -> Free rows**.
2. Open the main menu and choose **Edit -> Columns -> Free columns**.

To finish the example, print the set of marks and then save it in the **Practical** directory under the name **ExamMarks**.

			A	B	C	D	E	F	G
					Astrology	Palmistry	UFOlogy	Sorcery	Average
1					College of Irrational Sciences' Exam Results				
2					Astrology	Palmistry	UFOlogy	Sorcery	Average
3	Debra	De'Ath	Debra	De'Ath	25	49	64	40	44.5
4	Myra	Palmyra	Myra	Palmyra	89	91	62	78	80
5	Krystle	Ball	Krystle	Ball	70	35	55	62	55.5
6	Rose	Lee	Rose	Lee	71	11	60	49	47.75
7	Petra	Black	Petra	Black	68	71	38	51	57
8	Marie	Provan	Marie	Provan	52	42	68	91	63.25
9	Robert	Falconer	Robert	Falconer	37	31	34	50	38
10	Andreas	Colin	Andreas	Colin	7	12	8	47	18.5
11	Michael	Lamont	Michael	Lamont	42	119	60	49	67.5
12	Veronica	Rowout	Veronica	Rowout	68	84	95	89	84
13	Alison	Haywain	Alison	Haywain	51	61	56	56	56
14	Michael	Rooney	Michael	Rooney	52	78	36	41	51.75
15	Fred	Grant	Fred	Grant	49	82	28	16	43.75
16	Solomon	Rooney	Solomon	Rooney	73	66	59	81	69.75
17	Zuleika	Ball	Zuleika	Ball	46	65	51	48	52.5
18	Geoffrey	Quincy	Geoffrey	Quincy	51	17	40	58	41.5
19	Ebenezer	Rooney	Ebenezer	Rooney	70	55	42	39	51.5
20	Lucy	Borgia	Lucy	Borgia	76	40	36	71	55.75

Figure 3.3

Summary

In this example you have learned how to select blocks of cells and enter formulae into blocks with a single operation. You have also looked at locking and unlocking of rows and column.

Example 4 - Sorting

This example is about

- sorting spreadsheet records
- using sort key fields

The ability to sort blocks in a spreadsheet is often useful. For example, Hamish may want to print out two versions of the College mark spreadsheet:

One version is to be put up on the College notice board with the marks in alphabetical order of names.

On the other hand, the College governors prefer to see a record sheet printed in descending mark order, with the best student at the top.

Open the spreadsheet **ExamMarks** that you saved in your **Practical** directory in the last example.

In simple cases, sorting depends on the value of a key field in each set of data. A key field is the column or row which controls the order of the sorted records. For example, if Hamish wants to sort the results into order of overall marks, he will use column G as a key field.

Ex 4.1 Sorting

Sorting in **Schema2** is controlled by the Sort Dialogue, shown in Figure 4.1 below:

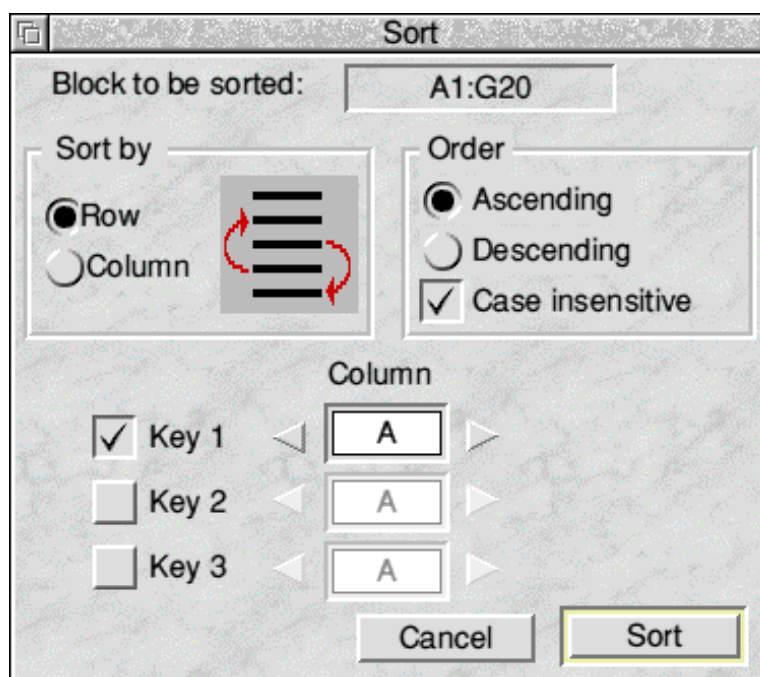


Figure 4.1

The box shows the block to be sorted and offers Hamish several simple choices:

Sort by Row or Column

Sort by Row means that complete rows are shuffled about, but each data item stays in its own column. Similarly **Sort by Column** means that columns are moved around. For the exam marks Hamish needs to shuffle the rows about depending on the values in column G, so that in principle he should select **Sort by Row**. In practice he need do nothing as the selected block is longer than it is wide, and Sort by Row is already picked as the default.

Ascending/Descending order

Ascending order means that when the sort is complete the entries in the key field will increase from the top down or from left to right. **Descending order** means the opposite. In this case the student with the highest average mark should come at the top; so Hamish must select **Descending order**.

Case insensitive

This only matters when a block is being sorted alphabetically, such as names. If a sort is **case sensitive**, it means that lower case letters such as **a b c** are treated as different from their upper case equivalents **A B C**. In most cases, the sort process does not need to be case sensitive. For sorting marks it doesn't matter, either way will do.

Finally, the dialogue includes up to three different key fields. If you need more than one they can be switched on one by one. In most cases you will only need to use the first one.

Each key field section of the dialogue has a central writable icon and two arrow icons, one at each side. At first, the icon shows **A**, since this is the first column on the selected block. Hamish must specify column **G** as the key field. One way Hamish can do this is to delete the **A** and change it to a **G**, using the mouse to place the cursor. Alternatively he can click on the right-pointing arrow 5 times, moving one column forward each time. If he goes too far he can use the left-pointing arrow to go back.

When the dialogue has been set up click on **Sort** to start the actual sorting process. It may take some time if a large amount of data is being processed.

To summarise, this is what Hamish does to sort the results. Do the same:

1. *Select a block which includes all the spreadsheet except the first two rows. To do this, click **Select** on the neutral area and then click **Adjust** on cell A3. (You don't want to sort the top two rows, since they hold fixed information; leave them out of the selected block.)*
2. *From the main menu choose **Utilities -> Sort** to open the Sort dialogue box, or you can press the **F6** key, the keyboard short cut for sorting.*
3. *Select the following options in the Sort dialogue box:*

Sort by Row
Descending order
Case insensitive
First key field: G

4. *Click on Sort. The records are now sorted into the order you want.*

Hamish prints out the spreadsheet in this form for the College Governors.

	A	B	C	D	E	F	G
1			College of Irrational Sciences' Exam Results				
2			Astrology	Palmistry	UFOlogy	Sorcery	Average
3	Veronica	Rowout	68	84	95	89	84
4	Myra	Palmyra	89	91	62	78	80
5	Solomon	Rooney	73	66	59	81	69.75
6	Michael	Lamont	42	119	60	49	67.5
7	Marie	Provan	52	42	68	91	63.25
8	Petra	Black	68	71	38	51	57
9	Alison	Haywain	51	61	56	56	56
10	Lucy	Borgia	76	40	36	71	55.75
11	Krystle	Ball	70	35	55	62	55.5
12	Zuleika	Ball	46	65	51	48	52.5
13	Michael	Rooney	52	78	36	41	51.75
14	Ebenezer	Rooney	70	55	42	39	51.5
15	Rose	Lee	71	11	60	49	47.75
16	Debra	De'Ath	25	49	64	40	44.5
17	Fred	Grant	49	82	28	16	43.75
18	Geoffrey	Quincy	51	17	40	58	41.5
19	Robert	Falconer	37	31	34	50	38
20	Andreas	Colin	7	12	8	47	18.5

Figure 4.2

Next, Hamish needs to rearrange the records in alphabetical order of the students' names, using 'phone book' order.

The surnames in column **B** form the main or **primary** key field, but they cannot be used to separate students with the same surname: for this we need to use the first names in column **A** as well. This column forms the **secondary** key field, used to determine the order of students who have the **same** surname.

The actual process of sorting by name is quite similar to sorting by marks. Follow these steps:

1. Select a block which includes all the spreadsheet except the first two rows (**A3..G20**).
2. Bring up the main menu and choose **Utilities -> Sort**, or press **F6**. The Sort dialogue box appears.
3. Select the following options in the Sort dialogue box:

Sort by row
Ascending order
Case insensitive
Key 1: **B**
Key 2: **A**

4. Click **SELECT** on **Sort**. The records are now sorted with the students listed in alphabetical order.

Look carefully at the resulting spreadsheet, which is shown in Figure 4.3. In particular, check that Schema2 has sorted the members of the Ball and Rooney families correctly.

	A	B	C	D	E	F	G
1			College of Irrational Sciences' Exam Results				
2			Astrology	Palmistry	UFOlogy	Sorcery	Average
3	Krystle	Ball	70	35	55	62	55.5
4	Zuleika	Ball	46	65	51	48	52.5
5	Petra	Black	68	71	38	51	57
6	Lucy	Borgia	76	40	36	71	55.75
7	Andreas	Colin	7	12	8	47	18.5
8	Debra	De'Ath	25	49	64	40	44.5
9	Robert	Falconer	37	31	34	50	38
10	Fred	Grant	49	82	28	16	43.75
11	Alison	Haywain	51	61	56	56	56
12	Michael	Lamont	42	119	60	49	67.5
13	Rose	Lee	71	11	60	49	47.75
14	Myra	Palmyra	89	91	62	78	80
15	Marie	Provan	52	42	68	91	63.25
16	Geoffrey	Quincy	51	17	40	58	41.5
17	Ebenezer	Rooney	70	55	42	39	51.5
18	Michael	Rooney	52	78	36	41	51.75
19	Solomon	Rooney	73	66	59	81	69.75
20	Veronica	Rowout	68	84	95	89	84

Figure 4.3

Some people find the idea of primary and secondary keys a bit awkward to grasp. You might find it worthwhile sorting the sheet in various different ways (perhaps exchanging the keys) and seeing what happens.

Summary

In this example you have learnt how to sort spreadsheet records and to use multiple sort key fields.

Example 5 - Growing your savings

This example discusses:

- changing the default spreadsheet size
- changing the number format
- using the absolute reference symbol @
- using the Go to cell command
- producing a basic graph

For many years Dr. Curser has been saving up for a trip to the North Pole to see the flying reindeer. Every month he puts another £10 note into an old sock he keeps under his bed, and he now has £850 put away. One day he receives a letter from the Effshort Building Society explaining that if he keeps his money there, he will be paid interest every month.

Greatly excited, Dr. Curser rushes up to the computer room where Hamish is still printing his exam results. Elbowing his colleague aside, Dr. Curser opens a new spreadsheet and uses it to predict how his savings will grow, both with the Building Society and in the sock under his bed.

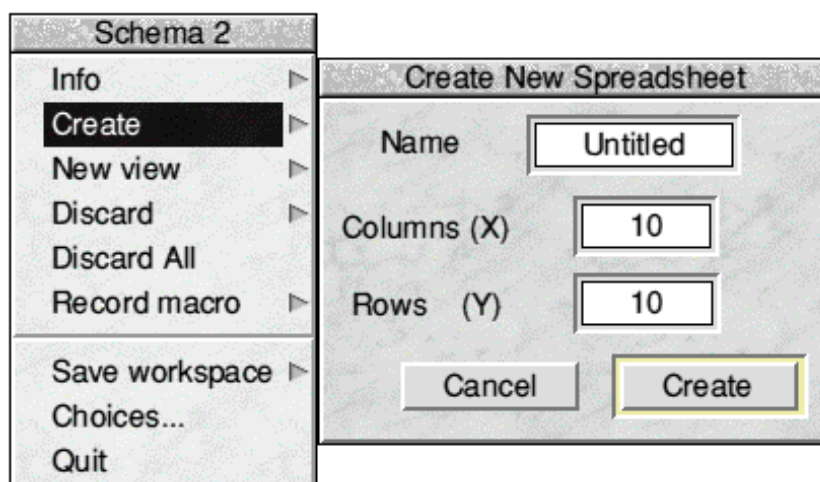
Dr. Curser needs to work out the state of his account each month for 10 years. Taking row titles into account, he needs 121 columns across the spreadsheet. In the other dimension, 5 rows are enough for his needs. (He could equally well have used a sheet with 5 columns and 121 rows, but he prefers growing things horizontally).

Ex 5.1 Changing the dimensions of a sheet

If you click on the Schema2 icon to open a new spreadsheet the height and width are set to default values, normally 10 columns by 28 rows. You can change this to any size you like, up to 30000 rows or 30000 columns - although the total size of spreadsheet you can handle is set by the amount of memory in your machine, and is very much less than 30000 x 30000 cells!

Set up Dr. Curser's new sheet with 5 rows and 121 columns:

1. Click **MENU** on the **Schema2** bar icon and choose **Create** from the menu.
2. Delete the default setting in the **Columns** box, and enter **121** as a new number of columns:.
3. Similarly, delete the default setting in the **rows** box, and enter **5** as the number of rows.
4. Click **SELECT** on the **Create** button. The dialogue box disappears and a new spreadsheet of the right size is created.



When a sheet has more than 26 columns, the column names start using two letters. Scroll the spreadsheet to the left. You will see that after column Z, the labels go on to AA, AB, ... AZ, then to BA, BB, BZ, and so on. The 121st column name is DQ.

Now scroll the spreadsheet back so that you can see cell AI.

As always, Dr. Curser began by planning the layout of the spreadsheet.

- He used column **A** for row labels.
- Dr. Curser was interested in a 10-year or 120-month period. He decided to number the months from 1 (in column **B**) to 120 (in column **DQ**). The month numbers appear in row **1**.
- Row **2** will show his total savings, month by month, on the assumption that he put a set amount into his bedsock every month. It starts with £850 in month 1 (which he has already saved), and goes up by the amount he saves each month.
- Row **3** represents his total savings if he puts all his money in the Building Society. It starts with an initial deposit of £850 in month 1 and goes up by the amount he pays in each month, but he also benefits from interest, on his existing savings, at the monthly rate.
- Dr. Curser decided to use cell **B4** for the current rate of interest (as a decimal, so that 11% will be shown as 0.11), and cell **B5** for the amount he saved each month.

Fill in the row titles in column **A**, as follows:

1. Select cell **A1** and type **Month number**. Press **Return** to enter the title and select **A2**.
2. Type in the following titles:

Bedsock total	<i>in cell A2</i>
Building Society total	<i>in cell A3</i>
Interest rate	<i>in cell A4</i>
Amount saved monthly	<i>in cell A5</i>

3. Double-click in the heading for column **A**. This widens the column so that all the item fits comfortably without overflowing into column **B**.

Ex 5.2 Generating column headings

Next, Dr. Curser fills row 1 with the month numbers from 1 to 120. He doesn't need to type each one; all he does is to enter the first number, and then make an automatic **Fill in** entry for the rest of the row. Here is exactly how to do it:

1. Select cell **B1** and enter the number 1.
2. Select block **B1..DQ1**. The easiest way to do this is:
 - Click **SELECT** on 1 in the side margin to select the whole row
 - Click **ADJUST** on cell **B1**.
3. Bring up the main menu and choose **Utilities -> Macros -> Fill in**.

Ex 5.3 Absolute References

Row 2 is supposed to hold the bedsock total. The first entry in **B2** is the amount Dr. Curser starts with - £850. Each subsequent month it increases by the amount of the monthly saving, which Dr. Curser plans to put into cell **B5**. Won't it always be £10? Perhaps not; Dr. Curser hopes for a salary rise one day which will let him save more.

The formula we need to enter into each of the cells in row 2 (apart from **A2** and **B2**) is one which takes the amount in the previous month and adds the contents of cell **B5**. In particular:

in C2 :	=B2+B5	
in D2 :	=C2+B5	
in E2 :	=D2+B5	and so on.

Clearly, we cannot use a straightforward block entry for these formulae; the **B5** in the first cell would be 'adjusted' to **C5** in the second cell, **D5** in the third, and so on - which we don't want. Every formula must refer explicitly to **B5**.

We could of course enter all 119 formulae by hand, but life is too short for such a clumsy procedure. Instead, we use a block entry with an absolute reference. This is just a way of ensuring that a cell name in a formula is **not** adjusted when it is filled in to different cells in the block. To make a reference absolute we put the **@** sign

in front of each of the row and column numbers of the cell involved : **@B@5**. We could also use the **\$** symbol, as in **\$B\$5**. It means the same as **@**.

Here is what to do:

1. Select cell **B2** and enter 850
2. Select cell **B5** and enter 10
3. Select block **C2 .. DQ2** and enter the formula
=B2 + @B@5 (Don't forget to press **Return**)

At this stage, the left side of your sheet should look like Figure 5.2:

	A	B	C	D	E
1	Month	1	2	3	4
2	Bedsock total	850	860	870	880
3	Building Society total				
4	Interest rate				
5	Amount saved monthly	10			

Figure 5.2

If you scroll up to column **DQ** you will see that by the end of 10 years Dr. Curser will have saved up a total of £2040.

Now Dr. Curser sets up row 3, which shows how much he can expect his Building Society account to be worth month by month.

The first entry, in cell **B3**, is the starting amount of £850.

Each subsequent entry is made up of three parts:

- The capital from the previous month is carried forward.
- A further saving (the amount in **B5**) is added.
- The Building Society adds interest on the previous capital, at the monthly interest rate (which is kept in **B4**).

The appropriate formulae for cells in row 3 are:

C3: **=B3 + B5 + B3*B4**
D3: **=C3 + B5 + C3*B4**
E3: **=D3 + B5 + D3*B4** and so on.

The right 'block' formula needs absolute references to **B4** and **B5**:

=B3 + @B@5 + B3*@B@4

To set up the row:

1. Select cell **B3** and enter 850.
2. Select cell **B4** and enter 0.009 (an interest rate of 0.9% per month).
3. Select block **C3..DQ3** and enter the formula:
=B3 + @B@5 + B3* @B@4

The left of the sheet is now shown in Figure 5.3:

	A	B	C	D	E
1	Month	1	2	3	4
2	Bedsock total	850	860	870	880
3	Building Society total	850	867.65	885.45885	903.42797
4	Interest rate	0.009			
5	Amount saved monthly	10			

Figure 5.3

and the far right of the sheet, after locking column A, is Figure 5.4:

		DN	DO	DP	DQ
1	Month	117	118	119	120
2	Bedsock total	2010	2020	2030	2040
3	Building Society total	4433.5816	4483.4838	4533.8352	4584.6397
4	Interest rate				
5	Amount saved monthly				

Figure 5.4

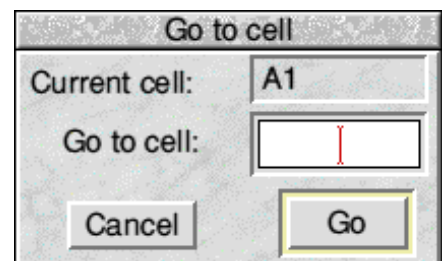
Dr. Curser is excited to see that the Building Society will make him a (relatively) rich man. Hamish smiles smugly and tells him that he (Hamish) is getting a rate of 27% **a year** with Sharkoid Leisure Investments Inc. How much can Dr. Curser expect at the end of 10 years if he also invests with Sharkoid? Change the interest rate and see for yourself.

Ex 5.4 The Go to cell dialogue box

When you have a large spreadsheet open like this one it can take a long time to scroll to the cell you want to see or select. In situations like this it is easier to use the Go to cell facility. This is a short cut for moving to a specified cell. For example, you can find cell **DQ2** directly by using the **Go to cell** dialogue box:

1. Click **MENU** over the body of the spreadsheet and choose **Utilities -> Go to cell**, or use the keyboard short cut **F5**. The **Go to cell** dialogue appears as shown on the right
2. Enter the name of the cell you want to see (in this case **DQ2**) and click on **Go** or press **Return**. The spreadsheet automatically scrolls and selects **DQ2**.

Figure 5.5



Ex 5.5 Number formats

Dr. Curser looks again at his sheet and notices that there is nothing to say that the contents of rows 2 and 3 represent sums of money. Some of the numbers show an absurd 'accuracy' of twelve decimal places.

Dr. Curser decides to apply a **format** to his numbers to make them look correct. Follow these steps:

1. Select block **B2..DQ3** (that is, most of 2 rows).
2. Bring up the main menu and choose **Effect -> Number Format**; alternatively type **F8** or click on the Format icon (shown) in the special icon bar. The system displays the format dialogue (Figure 5.6).
3. Click on the Currency button; set the number of decimal places to 2, and enter the £ sign into the Before number box. Then click on Apply.

Figure 5.6

The new format will be applied to all the numbers in the selected block.

We suggest you try some experiments with number formats. The sheet on the right (Figure 5.7) shows some of the different ways you could display the numbers 3.14159265 and 123456789:

Copy Figure 5.7 by using various types of number format.

	A	B
1	3.14159265	123456789
2	3	123,456,789
3	3.142	123.456.789,000
4	3.1416e+00	123 456 789
5	£3.14	28th January 0004
6	HK\$3.14	9.33pm
7	3.142 Km	1.235e+08

Figure 5.7

Ex 5.6 Graphs

Dr. Curser knows that pictures are often better than words, and decides to display his prospects in the form of a graph. He decides to plot both the bedsock and the Building Society totals against month number.

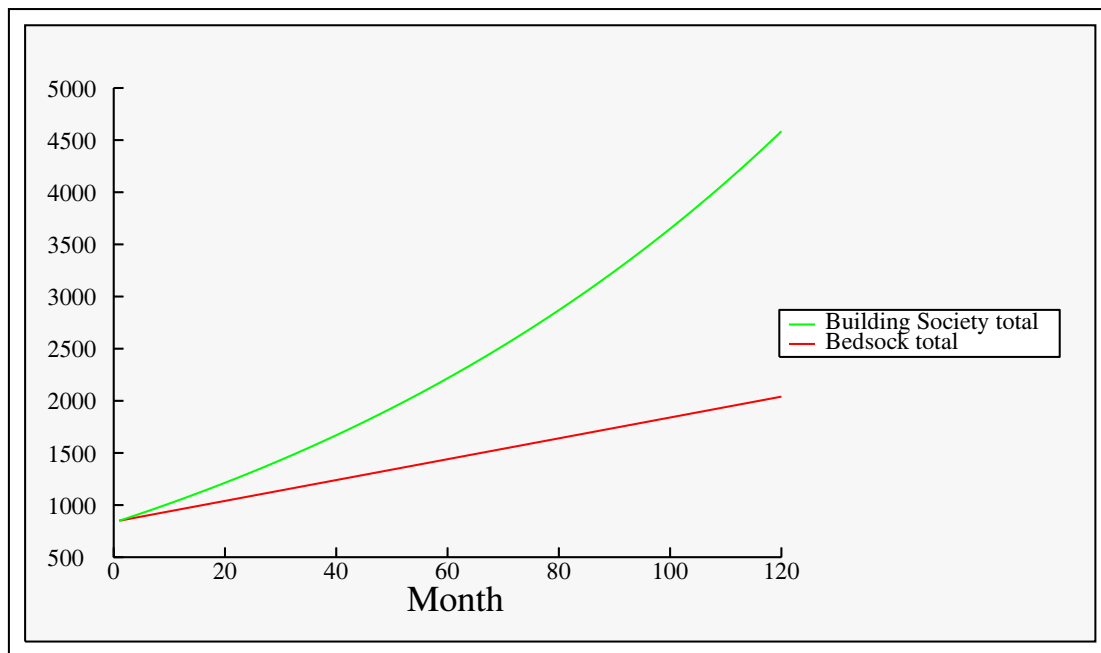
Schema2 offers many different types of graph, but the simplest to use is called **General**, which simply takes the data in the selected block and makes its own decision on how to plot it.

To plot a graph showing the growth of Dr. Curser's capital:

1. Select the whole of rows 1, 2 and 3.

2. Do one of the following:
 - Choose **Utilities** -> **Simple** graph in the main menu,
 - Click on the Simple graphs (right) icon in the icon ribbon, or
 - Use the keyboard short cut **Ctrl-T**.

Figure 5.8 shows the graph which you might obtain, depending on the interest rate you choose.



Ex 5.7 Mixed references

The @ sign can apply independently to the row and column of a cell reference. Sometimes it is useful to index along a row and not a column, or vice versa. Figure 5.9, for example, is a spreadsheet which displays the multiplication tables up to 10*10.

	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Figure 5.9

To create this spreadsheet:

1. Create a new 10 by 10 spreadsheet.
2. Fill row 1 and column 1 with the numbers 1 to 10.
3. Select **B2** and **Adjust** click on **J2** making the selected block **B2..J2**.
4. Enter the formula: **=@A2*B@1**

5. Click **SELECT** on the tick icon.

In cell J10 (for example) the formula is adjusted to **=A10*J1**

A copy of this spreadsheet is saved in the **Tutorial** directory as **Multiply**.

Summary

In this section we have covered a number of useful features for numerical spreadsheets:

- changing the default spreadsheet size
- changing the Number format
- using the absolute reference symbol (@)
- using the Go to cell facility
- producing a basic graph
- mixed references

Example 6 - Banners and Posters

This example looks at:

- effects -
 - fonts
 - character size
 - emphasis (bold and italic characters)
 - character and background colours
 - alignment
- cell borders and boxes
- column widths
- row heights
- big cells
- images
- magnification
- sheet borders and grid-lines
- protection

Every year the College of Irrational Studies organises a flag-day for the Shattered Crystal Ball Society, a charity which looks after distressed fortune-tellers. The academic staff, the students and the administration all act in friendly competition to see how much they can collect. The results are traditionally displayed in the form of a big poster.

This year, Hamish MacRihanish has volunteered to produce the results posters. He plans to make a basic design which can be used every year, with only minor alterations to the figures.

Happily the job can easily be done with Schema2, using the 'effects' feature to determine the appearance of the information in any cell. Effects are all called through the Effect submenu on the main menu, and each effect is applied to all the selected cells at the same time.

Many of the effects have keyboard short cuts.

Ex 6.1 Effects

Effects give you control over many aspects of material displayed in cells:

Text font - You can use any of the fonts installed on your RISC OS computer, as well as the standard or '**system**' font. Fonts have names such as **Trinity.Bold** or **Newhall.Oblique**.

Text size - You can choose any size of character within a wide range. Character size is measured in points, at 72 to the inch: so an 18-point typeface will be 0.25" high.

Text colour - You can change the colour of the characters using a 'colour picker' dialogue. If you plan to print the sheet, remember that a monochrome printer renders every colour except black and white as a shade of grey. What looks good on screen may be unreadable on paper.

Bold - You can specify that the contents of any cell is to be shown in bold type. For this to have any visible effect the font you choose for the cell must have a bold version; for example, you can embolden **Trinity** but not **System**.

The keyboard short cut for the **Bold** option is **Ctrl-B**.

Italic - Similarly, you can specify that the contents of any cell be shown in italic type. The font you choose for the cell must have an italic or oblique version. For example, you can italicise **Homerton** but not **Selwyn**.

The keyboard short cut for the **Italic** option is **Ctrl-I**.

Number format - This option displays a dialogue which lets you decide how to show numbers. We have already discussed this feature in Example 5.

The keyboard short cut to **Number format** is **F8**.

Alignment - This sets how the contents of each cell is aligned horizontally. In this list the keyboard short cut is given just after the type of alignment.

- **Left-alignment (Ctrl-F5)** puts the text up against the left edge of the cell.

- **Centre-alignment (Ctrl-F6)** puts the text in the (horizontal) centre
- **Right-alignment (Ctrl-F7)** places the text against the right edge.
- **Justify (Ctrl-F8)** : justification stretches text so that it reaches both edges of the cell. Justification is usually used with big cells (see Figure 6.3 later).

Vertical alignment - Vertical alignment is used to set the vertical position of text in cells. This is described in the section on section 6.3 (Big cells)

Graphics - This entry is used to control the way that pictures and graphs are drawn in cells.

You can find further details in section 6.4 (Images)

Cell Colour - You can choose the background colour of each cell. Don't make it the same as the text colour, or your text will be invisible.

Cell Border - Any cell or group of cells can be given a heavy black border on any or all of its four sides. This effect differs subtly from the others because it can treat the various cells in the selected block in different ways.

Cell borders are set up by this dialogue, reached by the Effect -> Cell borders menu entry (Figure 6.1).

To put a border anywhere on a single cell, select that cell, bring up the dialogue, click on one or more of the buttons labelled Above, Right, Below or Left, and then click on Set. Add will add new borders without taking away the old ones.

If you select Box, then the cell will have borders set on all four sides.

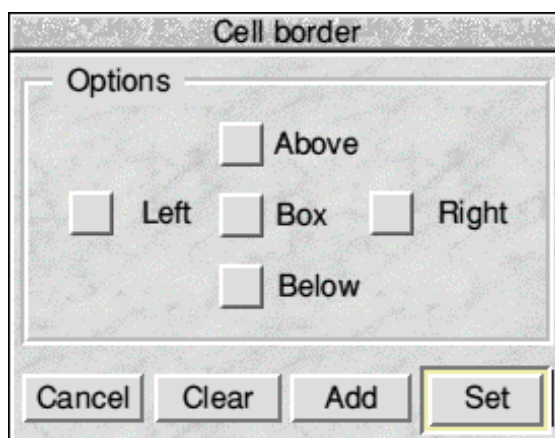


Figure 6.1

If more than one cell is selected, Above, Right, Below and Left still work the same way and put borders in each of the selected cells. However, Box only works on the cells at the **edges** of the selected block, and only marks the **outer** borders. This lets you put a single box round a block of cells of any size.

Ex 6.2 Column Widths and Row heights

There are various ways to adjust the width of a column or the height of a row. The simplest way to make a column wide enough for its contents is to *double-click* on its label in the top margin. The easiest way to make a row high enough is to *double-click* on the row label in the side margin.

You can explicitly set the dimensions of the rows and columns of the selection by choosing Edit -> Rows -> Height and Edit -> Columns -> Width from the main menu. Fill in the desired height or width and press **Return**.

A third way to change a row or column dimension is to place the mouse over the border between two of the margin cells (say between B and C in the top margin), watch for the pointer to change to a double-headed arrow, and drag the cell border one way or another. This takes a bit of practice, but is easy once you get the hang of it!

The first time Hamish made a poster with Schema2, he found it useful to experiment with effects and make a sheet like this one (Figure 6.2):

	A	B
1	Trinity.Bold 18-point	
2	<i>Corpus.Medium.Oblique 12-point</i>	
3	<i>Homerton Bold Oblique 12-point</i>	
4	Centre-Aligned	
5	Right-Aligned text	
6	A boxed cell	
7	A	boxed
8	group	of
9	cells	

Figure 6.2

Do the following. If the fonts we mention are not mounted on your machine, use any others you may have.

1. Open a new sheet, and fill cells **A1** to **A9** and **B7** to **B9** with text.
2. Select cell **A1**, and apply **Effect -> Bold** and **Effect -> Text size = 18**
3. Select cell **A2**, and apply **Text font -> Corpus** and **Effect -> Italic**
4. Select cell **A3** and apply **Text font -> Homerton.Bold.Oblique**
5. Select cell **A4** and apply **Alignment -> Centred**.
6. Select cell **A5** and apply **Alignment -> Right**.
7. Select cell **A6** and apply **Cell Border -> Box**.
8. Select cells **A7..B9** and apply **Cell Border -> Box**.
9. Select cells **A7..A9** and apply **Alignment -> Right**.
10. Double-click on top margin cell **A** to widen column **A**.
11. Double-click on side margin cells **1** & **3** to raise the heights of these rows.
12. Now change some of the colours (we can't show you examples in this printed manual).

Ex 6.3 Big Cells

You may often want to place a large item into a sheet - so large that it covers more than one cell horizontally, vertically, or both. To do this, select a block of cells of the right shape, and choose **Edit -> Big cell**. The block should either be empty, or contain at most only one cell containing data.

Once a big cell has been created, it obscures all the cells under it. If the block was empty the big cell takes the name of the cell at its top left-hand corner, otherwise it takes the name of the (one) cell which was filled.

To return a big cell to its normal status, select it and click on **Edit -> Big cell**.

If you enter a long string into a big cell, it will be split across several lines, with the breaks coming between words. If you include the sequence **IM** or **Im** in the string (I is the vertical bar character), it will be interpreted as a forced newline; this can be used to separate paragraphs.

If you select justification as the type of alignment for a big cell, each line will be stretched to line up with both sides of the cell.

Text in big cells can be aligned vertically using the entries in the **Effect -> V.Alignment** submenu:

- **Bottom** aligns the last line of the text with the lower edge of the cell.

- **Middle** aligns the middle of the text and the middle of the cell.
- **Top** aligns the first line of the text with the upper edge of the cell.

The various types of text alignment are shown below:

	A	B	C
1	This big cell uses left alignment, which means that the lines are lined up on the left		
2			
3	This cell uses right-alignment, which brings the right hand ends of the lines together.		
4			
5	The text in this cell is centred. Each line has the same amount of space on both sides.		
6			
7	Use of forced newlines This cell offers a demonstration of the way that newlines can be forced into strings displayed in big cells.		
8			
9	To make a new line use the vertical bar () followed by M or m.		

Figure 6.3

Ex 6.4 Images

Schema2 lets you plant a Drawfile into a cell. To do this, select a cell or a big cell, and drag the file icon of the Drawfile to the spreadsheet.

The way the Drawfile or image is positioned in the cell depends on the settings of the buttons in the Graphic display dialogue (figure 6.4), which is opened by the Graphics entry in the effects menu.

- If **Fit to cell** is **on**, the image is reduced or magnified in size so that it fits the cell exactly. If this button is **off**, the image is left unaltered in scale and aligned to the cell borders in the same way as text would be. Only part of the image may be visible,
- If **Lock aspect** is **on**, the image is not distorted - in other words, both vertical and horizontal magnification are done to the same scale. This generally leaves part of the cell blank unless you adjust the size of the cell to fit the image exactly.

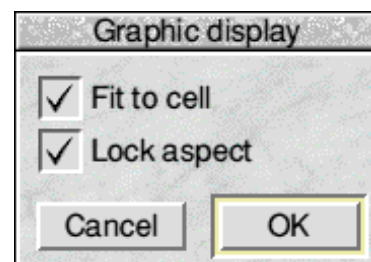


Figure 6.4

If the **Lock Aspect** button is off, and the **Fit to cell** button is **on**, the image is scaled independently on each axis, so that people in the picture may appear unusually fat or thin depending on the shape of the cell holding the image.

To put a large image into your sheet, mark a group of cells to be the size of the graphic you want to display and press **Ctrl F (Big cell)**. Your image is then automatically scaled to fill the big cell when you drop it on the sheet.

In his poster, Hamish plans to use the College logo, which you will find under the name of **CIS** in directory **Tutorial**. It is shown in Figure 6.5:

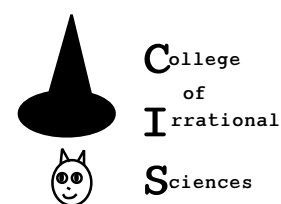


Figure 6.5

Ex 6.5 Scaling

You can display a spreadsheet at any scale you like (within limits). For example, you can blow up the sheet to examine fine details or shrink it to see all of (or more of) a large sheet. Note that scaling only applies to the on screen view of the sheet and does not alter the values for font sizes stored in the sheet.

Scaling is controlled from the Scale view dialogue, which is reached by choosing **View -> Scale view** in the main sheet menu. You can either change the scale directly to anything between 25% and 200%, or choose one of four useful values.

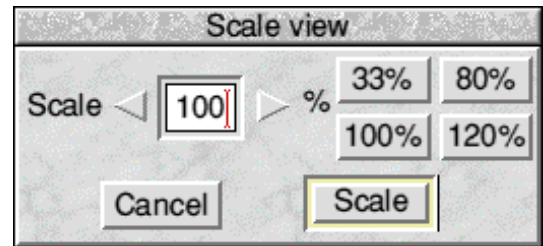


Figure 6.6

You can call up the the Scale view dialogue using **F11** as a keyboard short cut.

Ex.6.6 Borders and Grid lines

It is often useful to hide the sheet margins and the grid which separates the cells. You can do this by choosing **View -> Labels** from the main sheet menu to get rid of the margins, and (independently) **View -> Grid** for the cell grid. Second clicks on these entries will restore the hidden parts.

The keyboard short cuts for these features are:

Borders on/off:	Ctrl-O (letter Oh)
Grid on/off	F10

To start his poster, Hamish draws out a sketch like the one in Figure 6.7:

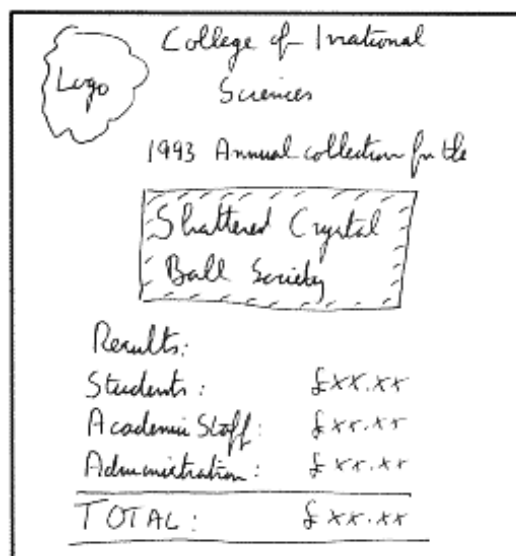


Figure 6.7

He also wants to include a note, as follows: "Particular thanks are due to Geoffrey Quincy for his wonderful work in raising £45.67, through his sponsored all night crystal ball gazing". He does the following:

1. Open a new sheet, of size 8 columns and 40 rows. This is about the right size to fill an A4 sheet. You may wish to reduce the magnification to 50% so that you can see the whole sheet.
2. Select the block **A1..B5**, menu and choose **Edit -> Big cell**.
3. Open the **Tutorial** directory. Drag the icon for the file (**CIS**) from the filer window on to the sheet. The image should appear in the top-left hand corner of the sheet.
4. Make **D2..G5** into a big cell and enter the text **College of Irrational Sciences**.
5. Apply the following effects to the big cell (**D2**):

Effect -> Text font -> **Homerton**
 Effect -> **Bold**

Effect -> Text size -> 32 pt
Effect -> Alignment -> Centre

At this point, Hamish has a spreadsheet like **PosterA** in the **Tutorial** directory.

6. Make **C11..G16** into a big cell and enter the text **Shattered Crystal Ball Society**.

7. Apply the following effects to the big cell (**C11**):

Effect -> Text font -> Homerton
Effect -> Bold
Effect -> Italic
Effect -> Text size -> Other -> 24 pt
Effect -> Alignment -> Centre
Effect -> V.Alignment -> Middle
Effect -> Cell border -> Box
Effect -> Cell colour -> Yellow

8. Fill in the amounts raised in **G23** for the Students, **G27** for the Academic staff and **G31** for the Administration. In **G35** enter the formula:

=G23 + G27 + G31

9. Apply Effect -> Number format -> Currency and enter '£' in the Before number box and set 2 decimal places.

10. Fill in the other text (except for the note about Geoffrey Quincy), using text size of 18pt.

The sheet should now look like **PosterB** in the **Tutorial** directory.

The last digit of the total is missing and the total needs some emphasising, so:

11. Double-click on the **G** at the top of column **G** to widen it.

12. Select the cell with the word 'Total' (**C35**). Apply Effect -> Bold.

13. Select **C35..G35**. Apply Effect -> Cell Border setting Above and Below.

14. Select **C35..G35**. Click **Menu** and choose Edit -> Rows -> Height. Change the height to 0.35 inches and press **Return**.

Now to add the note about Geoffrey Quincy ("Particular thanks are due to Geoffrey Quincy for his wonderful work in raising £45.67, through his sponsored all night crystal ball gazing"):

15. Select **C37..F39** and make into a big cell (**Ctrl F**). Enter the note.

16. Menu and choose View -> Grid and View -> Labels. Print the sheet by pressing the **Print** key and clicking on Print in the Print dialogue box.

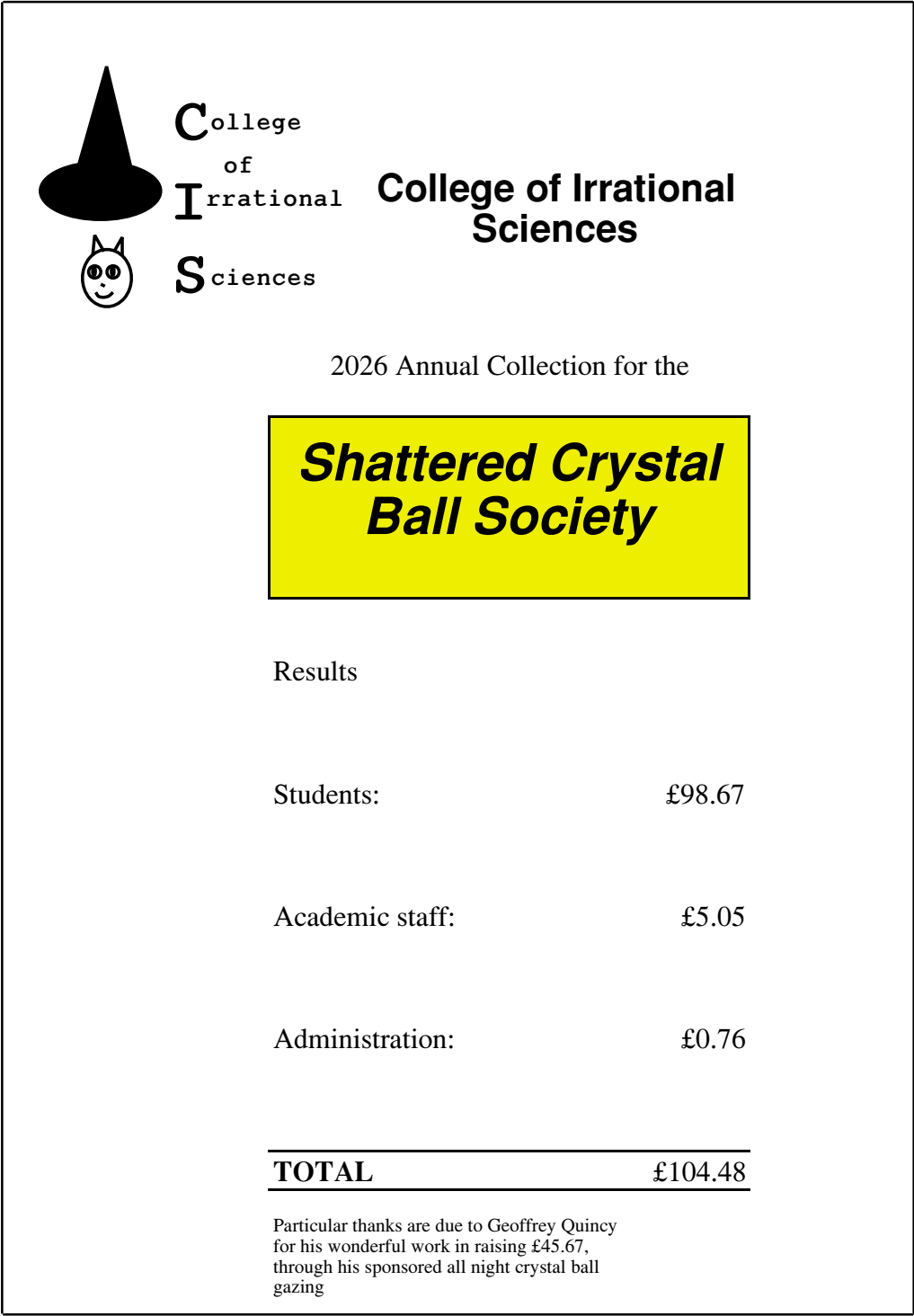


Figure 6.8

Next year, Hamish will be able to reload this sheet, change the year and the figures in cells **G23**, **G27** and **G31**, remove the note about Geoffrey Quincy and print the new version.

Ex 6.7 Protection

A few days later, Hamish realises that his poster might be damaged by someone who loads it, changes the design, and then saves the result, spoiling the original version. Fortunately **Schema2** has a protection system which will prevent anyone from making changes to a sheet, although it won't stop someone who is determined to make a change and knows what to do.

Each cell in the sheet has an invisible marker, which indicates whether the contents of the cell are eligible for protection. To set or clear the marker, choose **Utilities -> Protection** from the main menu.

If the protection marker is on, the system will not allow the cell contents to be changed. Anyone who wants to

alter the cell must first switch off the protection marker - a deliberate act which is unlikely to be done by mistake.

Hamish decides to protect the whole of his sheet except the cells used to hold the date, the individual totals and the grand total. Here is what he does when he has loaded the sheet:

1. *Select the whole sheet, and choose **Utilities -> Protection**. This **protects** every cell in the sheet.*
2. *Select **G23** and choose **Utilities -> Protection**. This **removes** the protection flag from that cell only ("unprotects" if you like jargon).*
3. *Do the same for cells **G27**, **G31** and **C37**.*

Summary

In this example we have shown how to change the appearance of items you put into cells. We have also looked at cell borders, the control of cell size, big cells, image handling, scaling and the options of hiding the cell grid and the label margins, and finally, protection.

Example 7 - Biorhythms

This example concentrates on:

- dates
- times
- temporal values
- functions

When Rona Toner first heard about Biorhythms, she was delighted; all the complex computations which go into making complex horoscopes could apparently be replaced by a series of simple calculations. Rona could work them out for herself, using Schema2. Here was a superb new candidate for the pantheon of Irrational Sciences!

Rona found an article on biorhythms, which read:

According to the theory of biorhythms, each one of us has three independent 'internal clocks' which start at the moment of birth and run without error to the ends of our lives.

- *The physical clock has a period or cycle time of 28 days- that is, our physical strength rises to a maximum, and falls again to a low value, every 28 days.*
- *The mental clock registers our mental powers and has a period of 45 days.*
- *The emotional clock is a measure of our emotional state and has a period of 62 days.*

At times when all three cycles are at maximum we are specially productive; similarly, when all the cycles are near the minimum, we tend to be slow and accident-prone.

Ex 7.1 Dates, times and temporal values

To set up a biorhythm chart for any person, Rona has to find out the subject's date of birth. All dates and times in Schema2 are handled as 'temporal variables'. A temporal value is a number which records the time which has passed since a given moment - actually the number of seconds since midnight on March 1st in the year 0. Clearly, any date and any time can be represented as a temporal variable; pre-Christian dates have negative values.

In the 20th Century temporal variables are rather large numbers, such as 62906860741. However, we hardly ever have to handle them in this form; Schema2 can accept data typed as normal dates or times, and there is a standard way to convert temporal variables to ordinary forms such as years, months, and so on.

The advantage of using temporal values instead of simple strings of characters is that you can do arithmetic with them; for example, you can easily work out the number of days between any two dates.

To get a date into a cell as a temporal, all you need do is to give that cell a **date** format, using the Number format dialogue. You can then select the cell and type a date such as

	22nd June 1936
or	25:12:1990
or	31/VIII/1812
or	17 nov 93

The system will work out the corresponding temporal value and store it in the selected cell.

As you can see by these examples, **Schema2** accepts a wide range of different date formats. If you leave off the two century digits, **Schema2** assumes that you mean the current century (so that **17 nov 93** is a date in **1993**). If you really want to enter a date in the first century **AD**, write it with two leading zeros as in **1.1.0067**.

The order of the date components is normally day-month-year, but can be changed if necessary.

If you type an impossible date (such as 32 December 1990 or 29 Feb 1900) Schema2 will give you a warning and then store the data as a string. This means you can't do arithmetic with it.

The presence of a date format in a cell means that its value will be assumed to be a temporal variable and displayed as a date. The form of the display may not be exactly the same as that used for input.

Ex 7.2 NOW()

Sometimes you need to know the date **today**- that is, the date when the spreadsheet is calculated.

The standard function **NOW()** can be used to load a cell with a temporal set to the time the cell is worked out. This time comes from the internal computer clock.

There are two ways to set this function into a cell:

- *Select a cell and type the characters **=NOW()** Don't forget the equals sign and the empty brackets. Select the tick icon.*
- *Select a cell and then click on the **Function menu** icon (the one marked **f()** in the special icon section of the input area). Then choose **Times** -> **NOW**. Finally select the tick icon.*

Ex 7.3 Other Standard Functions

Schema2 provides a good selection of standard functions other than **NOW()**. They are useful in many areas of application such as engineering, science, economics, mathematics and so on. Some of the functions are quite specialised, and you have to know the particular area fully to understand what they do. To make good use of Schema2 you don't need to know everything about all the functions. The ones you don't recognise can simply be left alone.

All the functions are fully described in the reference section of the manual. They are also listed in the Function menu under several headings, and the easiest way to include a function in an formula is to choose its entry in this menu.

Almost every function takes one or more **arguments** - that is, values it needs to use to calculate its answer. The arguments are placed in the pair of brackets which always follow the name of the function.

For example, there is a function, **SQRT**, which works out the square root of a number. What number? The number (or formula) in the brackets, like this:

=SQRT(9)

This is a valid formula which works out the square root of 9 and when calculated gives the answer: 3.

The argument does not have to be a number but can be any formula. When you enter a function name from the function menu the cursor is left between the brackets so that you can easily fill in the argument.

Functions can be parts of formulae, so that their results are combined with other numbers. For example, a valid formula is

=SQRT(A1+1) - A2

Rona can now start designing her biorhythm sheet. She decides to make the sheet work for all subjects, and calculate the strength of each of the three rhythms for the next 100 days. She uses the first two columns for labels and fixed data, such as the subject's name and birthday, so she needs 102 columns in all. She guesses (correctly) that 7 rows will be sufficient for the sheet.

Doing the detailed planning, Rona uses the top row for the subject's name. She will put the word **NAME** into **A1** and the actual name into **B1**.

The next row will hold the subject's birthday. **B1** will be the label, and **B2** the actual birthday, as a temporal.

The third row, starting in column **C**, will hold the subject's age in days. On the first day the age in **seconds** is given by the formula

(NOW() minus the birthday).

To convert this value to days she must divide by the number of seconds in a day - 24*60*60.

The fourth row will show the day number from the day the calculation starts.

The fifth row will show the subject's physical strength. We shall make it vary between 0 (minimum) and +2 (maximum) with an average of 1.

The daily physical strength can be worked out by the formula

$$=1 + \text{COS}(\text{time})$$

where the *time* depends in some way on the subject's age. How does it depend?

The cosine function **COS** takes degrees as its argument and completes a cycle (down and then up again) every 360 degrees. We want it to cycle round every 28 days. 360 degrees corresponds to 28 days, and the correct argument is therefore

$$(\text{age in days}) * 360/28.$$

D4, the first cell in the row (apart from the label cells), is set to the formula

$$=1+ \text{COS}(\text{C4}*360/28)$$

and the other cells, filled in by block entry, have corresponding formulae. We include the "1+" so that the result varies between 0 and 2, not between -1 and +1, as they would do if you used **COS** by itself.

The sixth and seventh rows, for mental and emotional strength, have similar formulae.

Once the planning is complete, here is what Rona does to build the biorhythm spreadsheet. Try it:

1. Open a new sheet and change its size to 102 columns by 7 rows.
2. Enter the following text into column 1:

A1 :	Name
A2:	Birthday
A3:	Age in days
A4:	Days into chart
A5:	Physical
A6:	Mental
A7:	Emotional

Double click on the column margin for column A to widen column A.

3. Enter your name into **B1**.
4. Select cell **B2** and apply Effect -> Number format -> Date.
5. Enter your birthday into cell **B2** using one of the formats shown earlier.
6. Enter this formula into **B3**:

$$=(\text{NOW}()-\text{B2})/(24*60*60)$$

and apply Number format -> Integer to the cell.

7. Into cell block **C3..CX3**, put the formula:

$$=\text{B3}+1$$

and apply Number format -> Integer to all this block.

8. Put 0 (zero) into **B4**
9. Into cell block **C4** to the end of the spreadsheet, enter the formula:

$$=\text{B4}+1$$

10. To block **B5..GX7** apply the Number format -> Fixed decimal with 2 places.
11. Into block **B5..GX5** enter the formula $=1+\text{COS}(\text{B3}*360/28)$
12. Into block **B6..GX6** enter the formula $=1+\text{COS}(\text{B3}*360/45)$
13. Into block **B7..GX7** enter the formula $=1+\text{COS}(\text{B3}*360/62)$

The left end of the spreadsheet Rona gets looks like this (Figure 7.1):

	A	B	C	D	E
1	Name	Rona Toner			
2	Birthday	1st April 1960			
3	Age in days	12275	12276	12277	12278
4	Days into chart	0	1	2	3
5	Physical	0.29	0.15	0.05	0.01
6	Mental	1.11	1.25	1.38	1.50
7	Emotional	1.99	2.00	2.00	1.99

Figure 7.1

(There is a copy of this spreadsheet saved as **Biorhythm** in the **Tutorial** directory).

The spreadsheet is not very informative, until Rona plots a graph of rows 4 to 7; then she gets Figure 7.2:

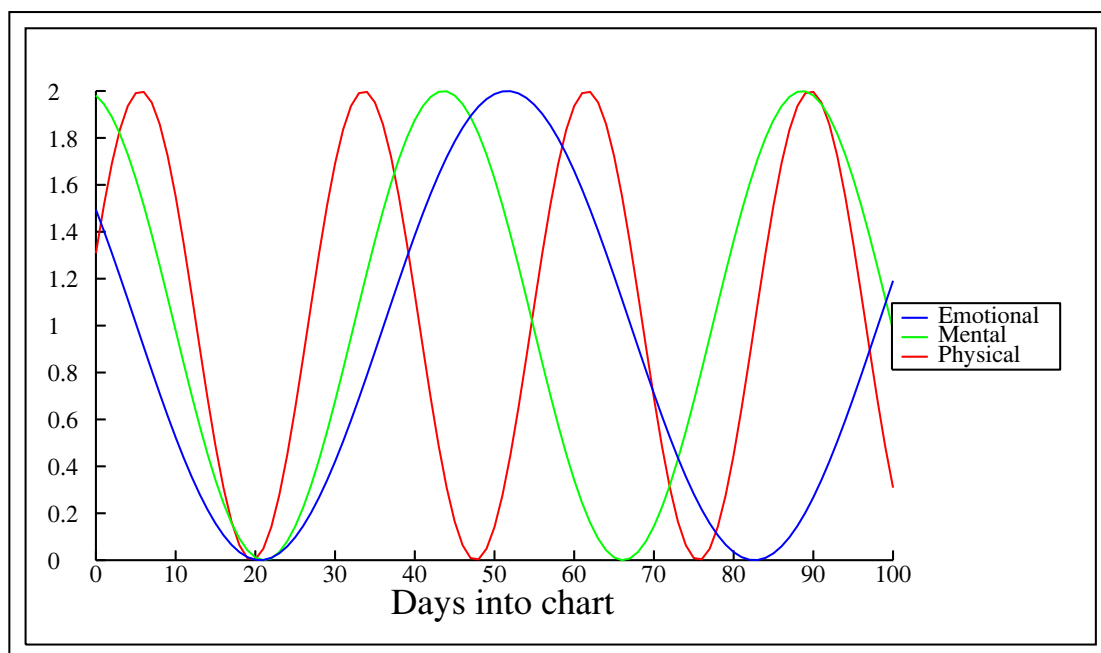


Figure 7.2

Plot the graph by dragging over the row labels 4, 5, 6 and 7 to select all of those rows and then click on the Simple graph icon in the icon ribbon.

According to this chart, Rona can expect a rotten time about 20 days into the prediction, when all three of her indicators are low. The chart itself is enough to make her feel bad!

As consolation, Rona remembers that she can use the sheet for anyone else; all they need do is to enter their name and birthday into cells **B1** and **B2**, and re-plot the graph.

Summary

In this example we have introduced you to dates and times, temporal values and functions.

Example 8 - Graphics

This example deals comprehensively with graphics. Points covered include:

- histograms
- pie-charts
- the graph dialogues:
 - New Graph
 - Selection
 - Options
- using non-contiguous data in graphs
- printing graphs
- exporting graphs

At one of the weekly Astrology tutorials, all the students had drawn up and handed in horoscopes of well-known men. The neatest and best was the chart for Hercule Poirot, prepared by Fred Grant.

"Excellent," said Rona after a glance, "you can see he was a typical Gemini."

"Actually, he was a Taurus," answered Fred. He went on with great daring, "I don't think people's birth signs have any effect on what they are like."

"You've no grounds for saying that. Where's your proof?" snapped Rona. She didn't like having her theories challenged.

Fred slowly pulled a creased sheet of paper from his jeans pocket. "I've spent a bit of time in the library," he said. "I looked up the birthdays and professions of a few hundred famous people, and they don't seem to cluster as you might expect."

Rona looked at the scrap of paper, which was covered in scribbles like

```

.....
Shakespeare      Writer      Aries
Shaw              Writer      Cancer
Shelley           Writer      Sagittarius
Sheridan          Writer      Gemini
Sibelius          Composer   Virgo
Smeaton           Engineer   Cancer
Snell             Scientist  Pisces
Suleiman          Astrologer Aries
.....
  
```

"You'll need to present your case better than that," Rona said. "You can have until next week."

Fred had his own computer, and a copy of Schema2. First, he entered all his findings into a long sheet, using three columns: Name, Profession and Sign of the Zodiac.

Then he sorted the entries, using the profession as a primary key and the sign as the secondary key.

This grouped the entries so that all the Virgo composers were together in one group, all the Capricorn scientists together in another, and so on. This made it easy to read off the number in

	A	B	C	D	E
1	Sign	Engineers	Writers	Composers	Astrologers
2	Aries	14	12	16	8
3	Taurus	15	20	15	9
4	Gemini	19	16	18	29
5	Cancer	21	15	17	8
6	Leo	13	19	21	5
7	Virgo	17	21	14	14
8	Libra	15	12	20	34
9	Scorpio	18	13	15	7
10	Sagittarius	17	15	14	13
11	Capricorn	13	20	16	9
12	Aquarius	20	14	12	37
13	Pisces	14	20	20	7

Figure 8.1

each group. (Later, Fred realised that he could have written a simple macro to do the same job, but at this stage he didn't know about macros.)

Fred then drew up a second sheet, like Figure 8.1 (You will find a copy of this sheet under the title Profession in directory Tutorial):

In this form the data certainly took up less space, but it was still just a jumble of numbers. To show their meaning, Fred tried plotting some graphs. First he selected the entire spreadsheet and called for a Simple graph plot. He got Figure 8.2:

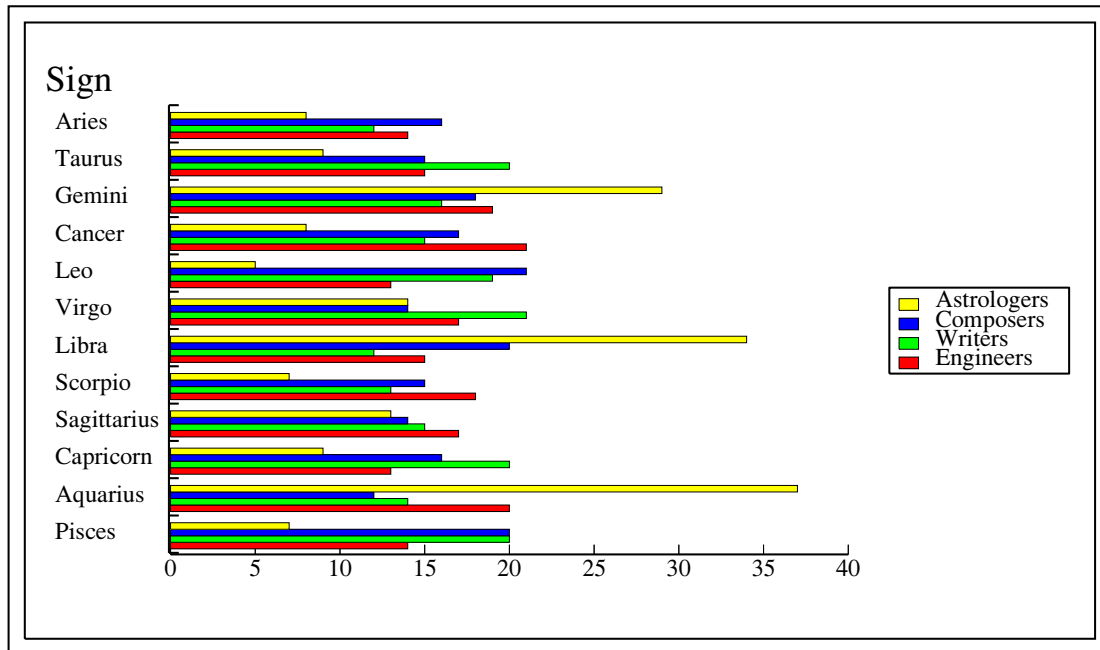


Figure 8.2

This graph is still too full of detail to carry a simple message, so Fred began to experiment with graphs which showed only part of the data.

Ex 8.1 The New Graph Dialogue

First, Fred plotted a histogram which showed the birth-sign only of engineers. To do this make sure that Profession is loaded, then:

1. Select columns A and B of the sheet.
2. Click Select on the Graphs icon on the icon ribbon
or
Open the main menu and choose Utilities -> Graphs,
This displays the dialogue shown below right (Figure 8.3)
3. Click on the menu button near the top of the dialogue. This brings up a pop-up menu of different graph types. Choose Vertical histogram.
4. Click on the Graph button.

The result of this action is shown below in Figure 8.4:

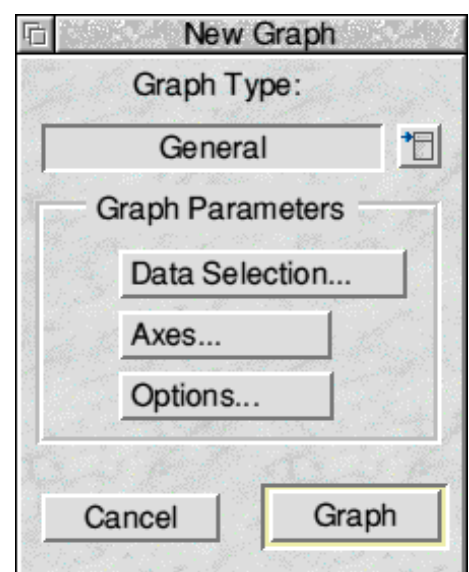


Figure 8.3

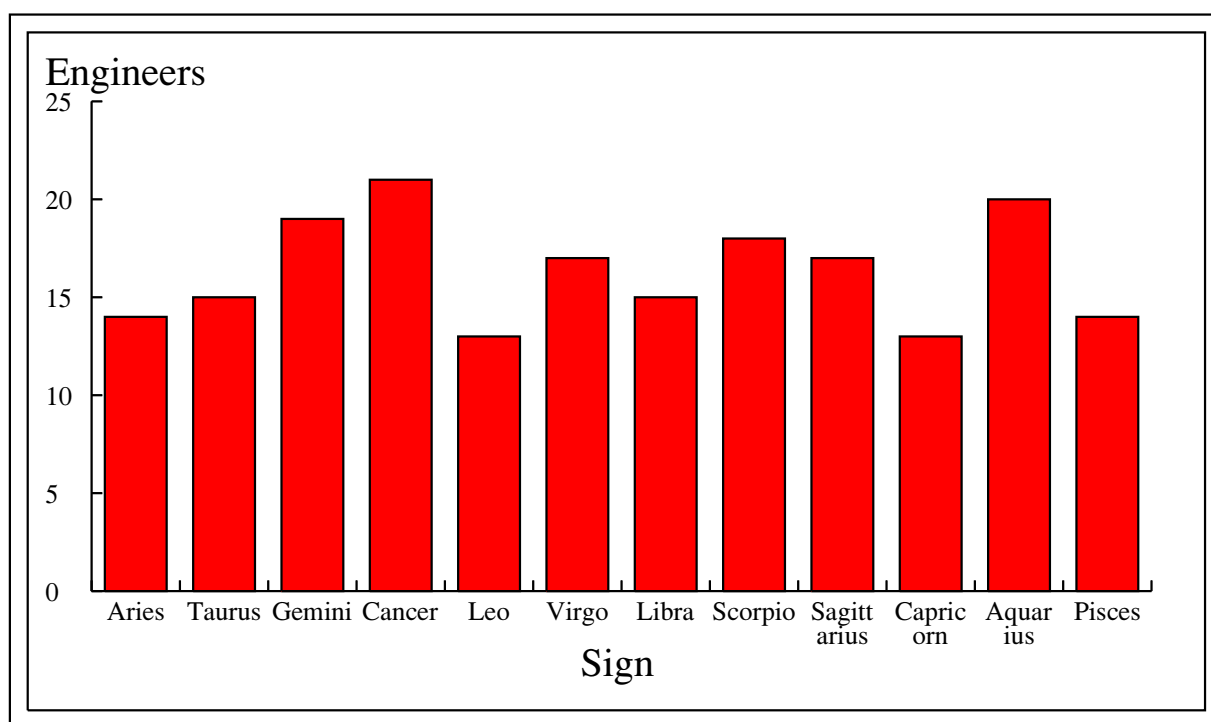


Figure 8.4

Ex 8.2 Horizontal histograms

Fred found the presentation of the vertical histogram a bit messy, especially as the names of some of the signs were split across two lines. As an alternative he tried a Horizontal histogram (Figure 8.5).

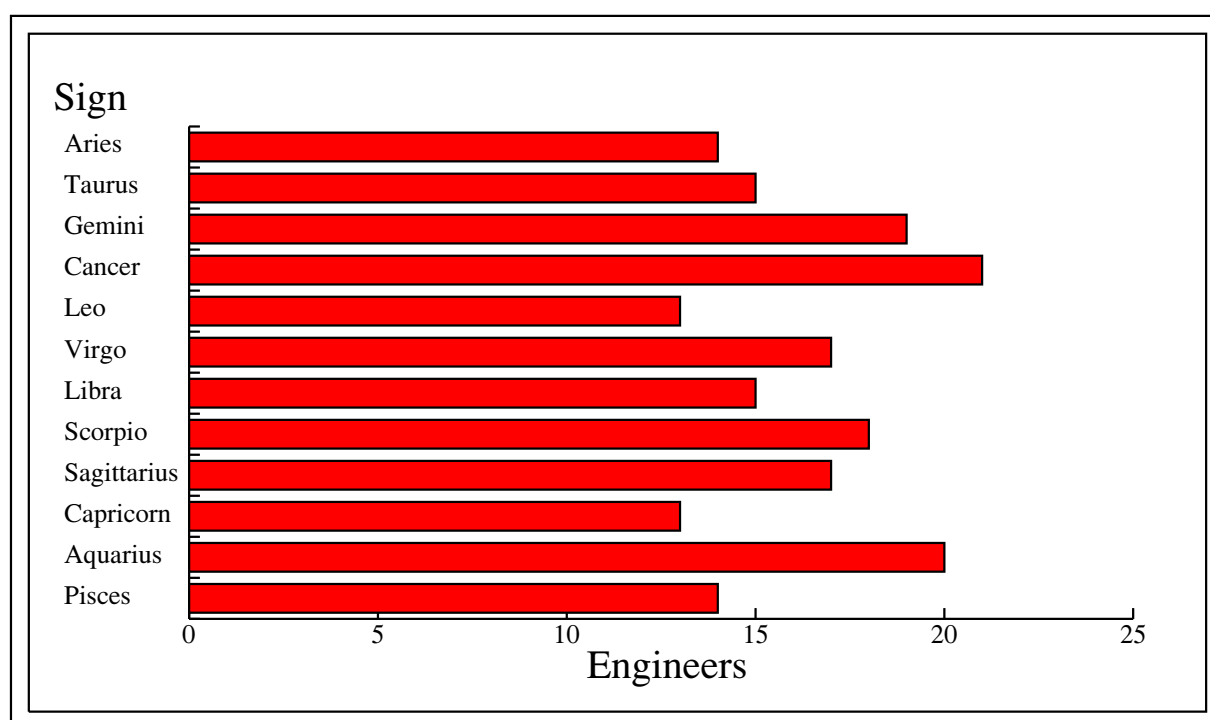


Figure 8.5

This layout seemed better to Fred than the upright histogram, so he decided to plot the other professions in the same way. However, he soon came across a small problem: the Schema2 graph plotter only works on a single selected block, whereas for all the professions (except the engineers) the column of sign names is separated from the column of numbers. For example, the list of star signs is in column A, but the numbers of astrologers are in column E.

Ex 8.3 The Data Selection Dialogue

This problem, where the data for the graph comes from separate areas of the sheet, often occurs. To help solve it, Schema2 has a Data Selection dialogue, which can be called from the New Graph dialogue.

To plot the Astrologers' histogram, take the following steps:

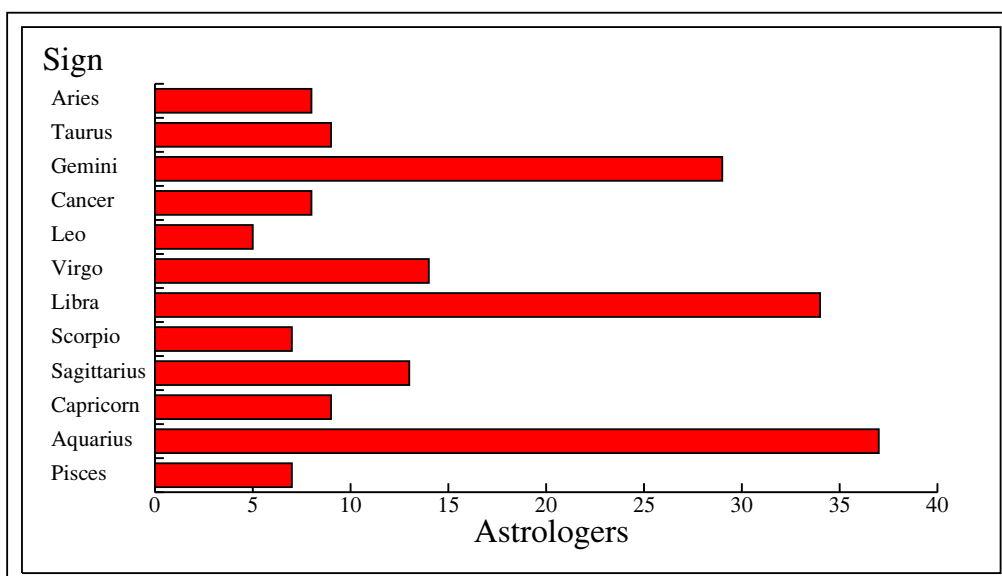
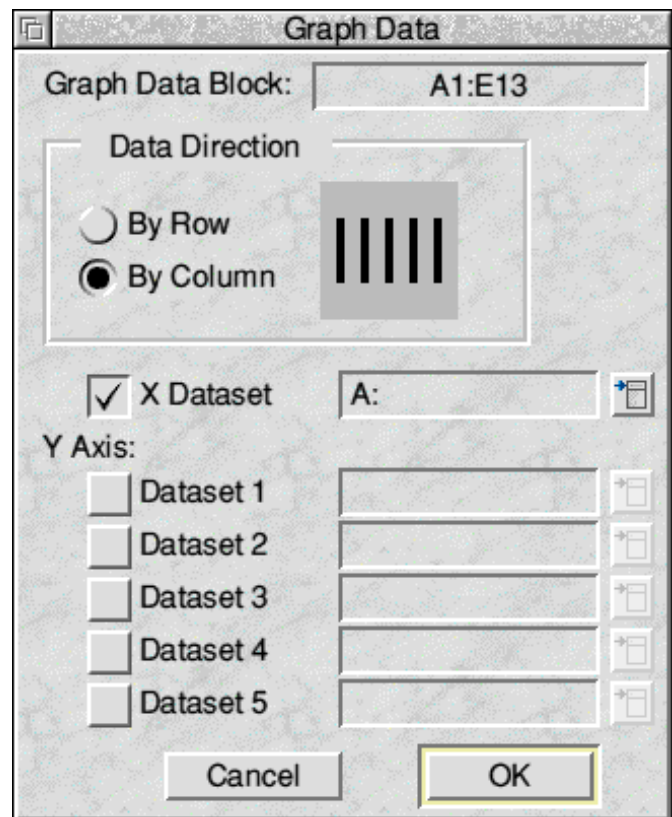
1. Select everything (**MENU** and choose Edit -> Select All). In general, you must select a block which holds all the data you want to plot, even if it also contains data to be ignored.
2. Click **MENU** over the sheet and choose Utilities -> Graphs.
3. Bring up the Graph type menu and choose Horizontal Histogram.
4. Click on the Data Selection button, on the New Graph dialogue. This brings up the Graph Data dialogue (Figure 8.6).

The X dataset represents the independent axis - which is another way of saying the bit you already knew before you collected any data. In this case, the independent axis is the list of star signs.

Each of the Y datasets represents a set of data which you have actually collected. In this case we only use one dataset, which is the number of astrologers born under each sign of the zodiac. The X dataset is already set correctly by default.

To set up Astrologers as the Y dataset:

5. Click on the Dataset 1 button. This automatically picks the first data column after the X dataset and enters its heading. However it also enables the menu button on the right.
6. Click on the menu button to the right of Engineers and choose Astrologers from the pop-up menu. This heading replaces Engineers in the window.
7. Click on the OK button in the Data Selection dialogue.
8. Click on the Graph button in the New Graph dialogue. This is the result:



Ex 8.4 The 3D effect

This graph carries a clear message, so Fred is keen to improve its appearance even further. One simple change is to use a three-dimensional effect.

1. Plot the graph again, and when the New Graph dialogue appears click on the Options button. This presents the Graph Options dialogue (Figure 8.8):

This lets you change the size of the graph and its orientation. The 3D button switches on a 'solid' effect which is believed to make graphs more vivid, though it doesn't actually add any real information.

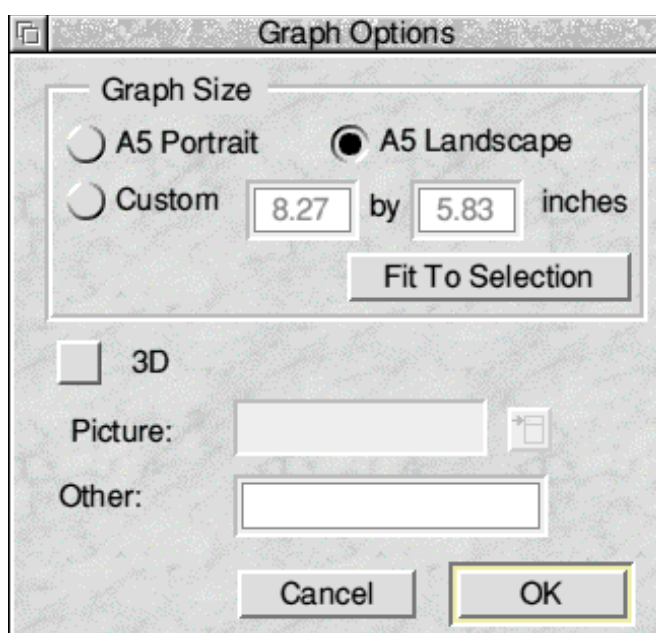


Figure 8.8

2. Click on the 3D button and then the OK button.
3. Click on the Graph button in the New Graph dialogue. The result is shown below:

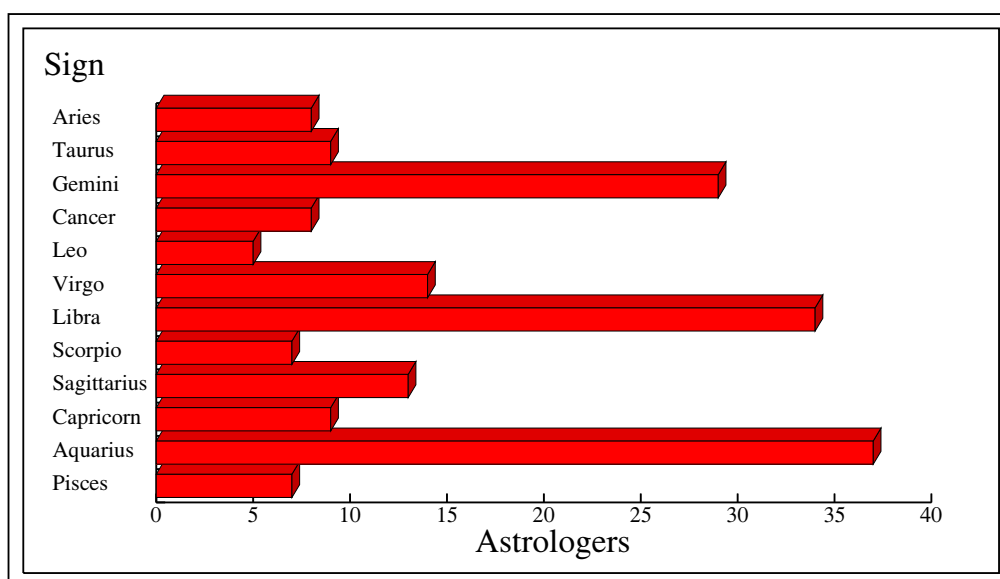


Figure 8.9

Ex 8.5 Pie-charts

Another way of displaying data is a pie-chart. To produce the chart shown in Figure 8.10 follow these steps (the chart looks even better in colour)

1. Select the whole sheet (columns A to E).
2. Bring up the New Graph dialogue.
3. Choose Pie chart as the graph type.
4. Open the Options dialogue and select 3D; then OK.
5. Click on Graph.

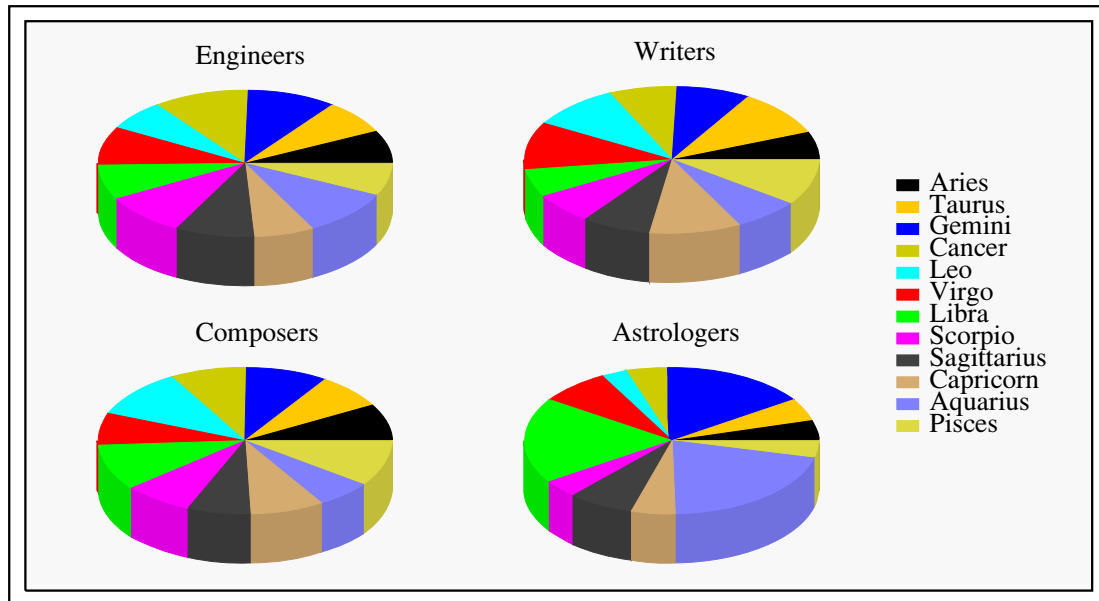


Figure 8.10

Ex 8.6 Printing graphs

When he had plotted his graphs Fred wanted to print them. This is easily done

1. Plot the graph on the screen.
2. Make sure the printer driver is loaded and the printer switched on.
3. Menu on the graph and select File -> Print.

Ex 8.7 Exporting graphs

A few days later Fred was writing up his case on a word processor and wanted to use some of the graphs as illustrations. This also proved very simple:

1. Put the word processor cursor at the place you want the graph.
2. Plot the graph on the screen in Schema 2.
3. Menu on the graph and choose the File -> Save dialogue.
4. Drag the file icon from the save dialogue to the word processor window.

Note: This method may not work with all word processors; it does work with Advance and Impression, and was used to write this manual.

Summary

This has shown how to plot, print and export graphs of various types

Example 9 - Decoding exam results

This example covers:

- adding and deleting rows and columns
- ranges
- array functions
- strings
- conditional formulae
 - hiding rows and columns

Hamish MacRihanish's exam result sheets, which we looked at in Examples 3 and 4, left the Board of Examiners underwhelmed. "In the old days," the Chairman said, "before we had these wretched computers, we were given the average mark for each **exam**. This helped us decide if any exam was unusually difficult, or exceptionally easy. Then there was no question of letting students know their actual marks - they were just told 'pass' or 'fail'. And there's another thing! All this use of the students' Christian names simply isn't professional. I want to see lists with names like K.Ball and F.Grant. Now see what you can do, my boy!"

During this tirade Hamish managed to jot down:

Exam averages
Hiding marks
Pass or Fail column
Compress names

Just as he was packing up his papers, Rona exclaimed, "Where's Egbert McSnoot? I know he's passed, but I can't see his name on the list!"

Scuttling out from the meeting, Hamish started repair work on his sheet.

Ex 9.1 Adding an extra row

Hamish reloaded his spreadsheet, **ExamMarks**, and decided to add an extra row to hold the average for each exam. The new row would be just below the subject titles, between the existing rows 2 and 3.

To add the row:

1. *Select the row immediately above the place you want the new row to be inserted - in this case, row 2.*
2. *Bring up the main menu and choose **Edit -> Rows -> Add rows**. When you see the writable field (box) with '1' in it, click the mouse **SELECT** button.*

The extra row is inserted and all the other rows move down. Data which was in row 5 is now in row 6, and similarly for all the other rows below the one you have just inserted. The last row in the sheet changes from 20 to 21.

If you examine the formulae in the cells you will see that all the references have been adjusted so that the sheet 'means' the same as it did before.

Ex 9.2 Ranges

Nearly every spreadsheet is used to add up complete rows or columns of figures. If a column or row is long, it can be tedious to enter each cell name one by one, and in any case it may lead to a very long formula. For example, Hamish was tempted to fill in the new cell C3, which is to hold the average mark for the Astrology exam, with the formula:

$$=(C4+C5+C6+C7+C8+C9+C10+C11+C12+C13+C14+C15+C16+C17+C18+C19+C20+C21)/18$$

However, Hamish was deterred by the thought that next year there may 150 students on the course. He remembered that Schema2 includes a feature called a *range*. This is a shorthand way of writing the names of all the cells in a given block.

To define a range, type in the names of the two cells at diagonally opposite corners, and separate the names by a pair of dots (..) which is called the block operator (or you can use the alternative block operator ><).

For example, the range **B4..D6** includes 9 cells:

B4 C4 D4
B5 C5 D5
B6 C6 D6

To specify a range you can write any two opposite corners, in any order; so that alternative ways of writing the same range are:

B4..D6, D6..B4, B6..D4 and D4..B6

Ranges can be used on single rows and columns, as in:

B2..E2 or A2..A30

Ex 9.3 Array Functions

The main use of a range is as the argument of an array function. An array function is one which calculates a single arithmetical result from a whole range of cells - such as their sum, which adds them all together, or their average.

To add up a range of numbers, use the formula:

=SUM (range)

For example:

=SUM (A2..B3)

gives the same result as:

=A2+A3+B2+B3

because the range **A2..B3** consists of the four cells **A2, B2, A3 and B3**.

You can now see that the long formula to work out the average Astrology mark can be replaced by:

=SUM(C4..C21)/18

An even better function for this purpose is **AVERAGE**, which calculates the (arithmetic) average value of the cells in its range. You can use this function without even having to count how many numbers there are in the block:

=AVERAGE(C4..C21)

The formulae in **D3, E3 and F3** are quite similar, and can all be filled in by a single block entry.

To add the 'average' cells:

1. Put the string **Average for paper** into cell **A3**
2. Select cells **C3..F3** and fill the block with the formula

=AVERAGE (C4..C21)

*If you like, you can get the function name from the function menu, choosing **Array -> AVERAGE**.*

3. Apply **Effect -> Number format -> Integer** to **C3..F3**.

The top left hand corner of your sheet now looks like this (Figure 9.1):

	A	B	C	D	E	F	G
1			College of Irrational Sciences' Exam Results				
2			Astrology	Palmistry	UFOlogy	Sorcery	Average
3	Average for paper		55	56	50	56	
4	Debra	De'Ath	25	49	64	40	44.5
5	Myra	Palmyra	89	91	62	78	80
6	Krystle	Ball	70	35	55	62	55.5
7	Rose	Lee	71	11	60	49	47.75
8	Petra	Black	68	71	38	51	57
9	Marie	Provan	52	42	68	91	63.25
10	Robert	Falconer	37	31	34	50	38

Figure 9.1

Ex 9.4 Adding and deleting rows in the middle of ranges.

Next, Hamish turns his attention to Egbert McSnoot, the student missing from the record. It turns out that he was re-sitting the exams for the fifth time, so it's not surprising he wasn't included! This time he's improved a lot; his marks for the four exams are 67, 78, 49 and 56, and his correct place in the list is immediately after Marie Provan.

Hamish adds a row after row 9, and types in the name and the marks. He refills the results column, by selecting block **G3** to **G22** and pressing **Return**.

Hamish does **not** have to adjust the formulae for the averages. In column **C**, the block reference **C3..C21** is already automatically changed to **C3..C22**, and the average is up to date.

All this happens whenever a row is inserted from the inside of a range.

This year no one has been caught cheating, so Hamish does not have to disqualify any students and remove their names from the list. If he did, he would need to select that student's row, open the main menu and click on **Edit -> Rows -> Delete rows**. The row would disappear from the sheet, all rows below would be renumbered and ranges spanning the deleted row would be adjusted automatically.

Everything we have said about adding and deleting rows applies equally to adding and deleting columns.

Ex 9.5 Strings

One of Hamish's tasks is replacing each student's full name by an initial and a surname. He can do this automatically using string operations and functions

The most common string operation in Schema2 is Cat, which stands for **concatenation**. It takes any two pieces of text and joins them together.

Cat is always used in formulae. To put a segment of text into a formula, you need to enclose it in double quotes, like this:

"A text"

A segment of text in quotes is often called a **string**.

To give a simple example, the formula:

= "Henry" cat "VIII" works out to **HenryVIII**

Similarly,

= "Isaac" cat "Newton" works out to **IsaacNewton**

If you want a space between the items, you must write

= "Isaac" cat " " cat "Newton" which gives Isaac Newton

(notice the space in quotes between the two **cats**).

Formulae with **cat** can use cell names instead of quoted strings. For example, if **A1** holds **Albert** and **B1** holds **Einstein**, the formula

=A1 cat " " cat B1 gives Albert Einstein

Note: In formulae **cat** must be followed by a space if the next character is a letter or decimal digit. For example, always type **cat B1**, not **catB1**.

A useful string function in Schema2 is **MID**. This function is used to extract any portion of a segment of text. The function requires three values, or **arguments**, to specify the text and which part it should extract. The first argument is the segment of text. The second and third arguments specify the first and last character numbers of the segment. The characters in the text are notionally numbered from 1 up. The arguments are enclosed by the brackets and separated from each other by commas.

To give some examples:

=MID("KEYBOARD",4,6) gives BOA

=MID("MONITOR",1,1) gives M

and

=MID(B4,1,1) gives the first character in B4

Hamish can now construct the list of names demanded by the Chairman.

Consider cells **A4** and **B4**, which hold the texts **Debra** and **De'Ath**. To squeeze up this name Hamish needs the first letter of the first name, a dot, and the whole of the second name: **D.De'Ath**. A suitable formula is:

= MID(A4,1,1) cat "." cat B4

Hamish decides to add an extra column (after column **B**) and use it for the squeezed names. He does this as follows:

1. Select column **B**, choose Edit -> Columns -> Add columns, in the main menu, follow the dialogue arrow, check that a box comes up with a '1' in it and press **Return**.
2. Select **C4..C22**, in the new column **C**, and enter the formula:

= MID(A4,1,1) cat "." cat B4

3. Double-click on the column's top margin to widen it.

The top of the spreadsheet now looks like this :

Figure 9.2

	A	B	C	D	E	F
1				College of Irrational Sciences' Exam		
2				Astrology	Palmistry	UFOlogy
3	Average for paper			56	57	50
4	Debra	De'Ath	D.De'Ath	25	49	64
5	Myra	Palmyra	M.Palmyra	89	91	62
6	Krystle	Ball	K.Ball	70	35	55
7	Rose	Lee	R.Lee	71	11	60
8	Petra	Black	P.Black	68	71	38
9	Marie	Provan	M.Provan	52	42	68
10	Egbert	McSnoot	E.McSnoot	67	78	49
11	Robert	Falconer	R.Falconer	37	31	34

Ex 9.6 Conditional Formulae

Conditional Formula is a technical name for the situation where you have two different rules for working something out. If one set of circumstances applies, you take the first rule; otherwise you take the second.

Hamish has been ordered to add a **"PASS/FAIL"** column to his spreadsheet. He needs a conditional formula to decide which of the two words "Pass" or "Fail" should appear in a cell.

The rule is that if a student scores an average of 50% or more, they pass; otherwise, they fail. We can express this rule in tabular form with a **condition**, which specifies the circumstance and two formulae: one to be applied if the condition is true, and one if it is false.

Condition:	Average is 50 or more
Rule if condition is true:	Enter "PASS"
Rule if condition is false:	Enter "FAIL"

To give another example, consider the College prize fund. Every year good students can expect to earn prizes worth £15 for each mark that their average exceeds 70%. Students with 70% or less get no prize. The rule is:

Condition:	Average = 70 or more
Rule if condition is true:	Prize = 15*(Average - 70)
Rule if condition is false:	Prize = 0

To be used in Schema2, a conditional rule must be written out in a form called a **conditional formula**:

A **condition** is a kind of formula which always evaluates to either **True** or **False**. Conditions always use one of the relational symbols

=	(equals)
<>	(not equal to)
>	(greater than)
<	(less than)
>=	(greater than or equal to)
<=	(less than or equal to)

The relational symbols link two quantities, one of which is usually a cell name. For example, if cell **H4** holds a student's exam marks, then the condition:

H4 >= 50

will be **true** if the student has actually scored 50 or more; otherwise (for example if he has scored 43) the condition is **false**.

In a Schema2 formula a conditional formula is written in three parts, separated by the keywords **if**, **then**, **else** and **endif**. The arrangement is as follows:

if condition then rule for true condition else rule for false condition endif

The conditional formula can be (and usually is) a complete formula on its own, when it would be preceded by an = sign. It can also be part of a larger formula

Hamish can now add a **Pass/Fail** column and a **prize** column to his sheet. The Pass/Fail column is labelled I, and the entry in **I4** (for example) should be **PASS** if **H4** is 50 or more; otherwise **FAIL**. Similarly the prize column is J; it should be blank unless a student wins a prize.

The right conditional formula for **I4** is

=if H4 >= 50 then "PASS" else "FAIL" endif

and for **J4**

=if H4 > 70 then INT(15 *(H4 - 70)) else "" endif

The function INT takes any number (which may be a fraction) and returns the nearest whole number below.

Hamish uses it to avoid odd pence in the prizes.

Note the double quotes which mean **Print nothing**. This ensures that the cell does not contain an error condition.

Here is what Hamish does to add the Pass/Fail and prize columns

1. In I1 enter the word **Result**, and in J1, the word **Prize**.
2. Select block I4..I22 and enter the formula
=if H4 >= 50 then "PASS" else "FAIL" endif
3. Select block J4..J22 and enter the formula
=if H4 > 70 then INT(15*(H4-70)) else "" endif

The result is shown in Figure 9.3.

	A	B	C	D	E	F	G	H	I	J
1				College of Irrational Sciences' Exam Results						
2				Astrology	Palmistry	UFOlogy	Sorcery	Average	Result	Prize
3	Average for paper			56	57	50	56			
4	Debra	De'Ath	D.De'Ath	25	49	64	40	44.5	FAIL	
5	Myra	Palmyra	M.Palmyra	89	91	62	78	80	PASS	150
6	Krystle	Ball	K.Ball	70	35	55	62	55.5	PASS	
7	Rose	Lee	R.Lee	71	11	60	49	47.75	FAIL	
8	Petra	Black	P.Black	68	71	38	51	57	PASS	
9	Marie	Provan	M.Provan	52	42	68	91	63.25	PASS	
10	Egbert	McSnoot	E.McSnoot	67	78	49	56	62.5	PASS	
11	Robert	Falconer	R.Falconer	37	31	34	50	38	FAIL	
12	Andreas	Colin	A.Colin	7	12	8	47	18.5	FAIL	
13	Michael	Lamont	M.Lamont	42	119	60	49	67.5	PASS	
14	Veronica	Rowout	V.Rowout	68	84	95	89	84	PASS	210
15	Alison	Haywain	A.Haywain	51	61	56	56	56	PASS	
16	Michael	Rooney	M.Rooney	52	78	36	41	51.75	PASS	
17	Fred	Grant	F.Grant	49	82	28	16	43.75	FAIL	
18	Solomon	Rooney	S.Rooney	73	66	59	81	69.75	PASS	
19	Zuleika	Ball	Z.Ball	46	65	51	48	52.5	PASS	
20	Geoffrey	Quincy	G.Quincy	51	17	40	58	41.5	FAIL	
21	Ebenezer	Rooney	E.Rooney	70	55	42	39	51.5	PASS	
22	Lucy	Borgia	L.Borgia	76	40	36	71	55.75	PASS	

Figure 9.3

Ex 9.7 Hiding Rows or Columns.

Figure 9.3 gives Hamish a sheet with a Pass/Fail results column, but the sheet still shows students' individual marks. The Board of Examiners want to publish a sheet with these items suppressed.

Hamish can do this by hiding certain rows and columns. If you hide part of a sheet it seems to disappear, and does not figure in any printed version of the sheet; but the information is not really lost and when you reveal hidden areas they simply reappear.

To produce a printed sheet with hidden rows and columns:

1. *Select columns A and B.*
2. *Bring up the main menu and choose Edit -> Columns -> Hide.*
3. *Select columns D to H.*
4. *Bring up the main menu and choose Edit -> Columns -> Hide.*
5. *Select row 1.*
6. *Bring up the main menu and choose Edit -> Rows -> Hide.*
7. *Turn off the grid and labels with View -> Grid and View -> Labels.*
8. *Print the sheet.*
9. *Bring up the main menu and choose Edit -> Row -> Reveal.*
10. *Bring up the main menu and choose Edit -> Column -> Reveal.*

The result is shown on the right in Figure 9.4.

Summary

In this example we have looked at adding and deleting rows, ranges, array and string functions, conditional formulae and the temporary concealment of rows and columns - all tools you need to produce an accurate sheet with a good appearance.

	Result	Prize
D.De'Ath	FAIL	150
M.Palmyra	PASS	
K.Ball	PASS	
R.Lee	FAIL	
P.Black	PASS	
M.Provan	PASS	210
E.McSnoot	PASS	
R.Falconer	FAIL	
A.Colin	FAIL	
M.Lamont	PASS	
V.Rowout	PASS	
A.Haywain	PASS	
M.Rooney	PASS	
F.Grant	FAIL	
S.Rooney	PASS	
Z.Ball	PASS	
G.Quincy	FAIL	
E.Rooney	PASS	
L.Borgia	PASS	

Figure 9.4

Example 10 - Financial predictions

This example takes you in detail through the process of planning and setting a cash flow projection for a new business. Unlike the other examples, we don't tell you exactly what to do: just follow the steps, using your own set of numbers if you prefer.

Myra Palmyra, one of the best students of her year, decides to set up as a fortune teller in a booth on the pier at Ullapool. She has no savings but believes the bank will give her an overdraft. She is keen to know if her business will be successful. She goes for advice to Hamish.

To Myra's surprise, Hamish begins by asking her lots of questions. Some of them are quite hard, and lead Myra to 'phone her bank, her magical trade suppliers, the pier management company, and various other fortune tellers all over the country. In the end, they assemble the following data:

Bank: Will allow Myra an overdraft up to £3000, with an interest rate of 2% per month.

Rent: The booth on the pier will cost Myra £250 per month, plus £500 'key money'.

Initial Expenses: Myra's crystal ball, teacups and other furniture will cost £500.

Monthly Expenses: Each month Myra will spend £50 on her telephone, on leaflets to hand to customers, and on cleaning and security services. Heating will cost her £25 a month during the summer months of April to September, and £50 a month throughout the winter.

Personal Pay: Each month Myra will need to draw out £250 for food, clothes and rent (by the terms of her lease she won't be allowed to sleep in her booth).

Fees: Myra discovered that most clients expect to pay £5 to have their fortunes told.

Number of customers: Myra found this aspect of her plans the hardest to predict. All the advice she got was that there would be more clients in the summer, with May, June and September being the best months. In the second year (which is as far as she was willing to look ahead) numbers would be up by 50% as existing clients recommended her services to new clients.

Myra planned to start in August (as soon as possible after finishing the course) and the numbers she came up with are listed below:

		First year	Second year
August	90	270	
September	180	270	
October	120	240	
November	80	150	
December	40	60	
January	20	60	
February	20	30	
March	30	30	
April	120	40	
May	280	180	
June	240	240	
July	270		300

The sheet which Myra and Hamish drew up is shown in Figure 10.1. You can also find it under the title CashFlow in directory Tutorial.

Most of the sheet was filled in with data from the answers to Hamish's questions. Formulae are used in the following places:

1. The total income row is found by multiplying the estimated number of clients by the unit charge. The block entry in cells B4..Y4 is $=B@3*B2$.
2. The bank interest is worked out as 2% of the previous month's overdraft (if any). If the balance is in credit no interest is credited.

The entry in this line uses a conditional formula:

In this case the condition is that the previous month's balance should be positive - that is, in credit. If it is, the interest is zero. If there is an overdraft, the interest (chargeable as an outgoing) is -2% of the balance.

The bank balance itself is recorded in the cells in row 14; so the typical entry in a cell in row 12 (which records the interest) uses the cell in row 14 from the previous month. The interest in the first month is zero, but the entry for C12 is:

$\text{=B14} > 0 \text{ then } 0 \text{ else } -0.02 * \text{B14} \text{ endif}$

This entry is repeated to the end of the sheet.

3. *The total outgoings each month are recorded as the sum of individual outgoings eg rent, key money, initial expenses, monthly expenses, heating, pay and bank interest. So B13 has the formula:*

=SUM(B6..B12)

4. *For most months the balance is calculated by taking the previous balance, adding the income and taking away the outgoings. However, for the first month there is no previous balance, so we compute the balance at the end of this month as the first month's income, less the first month's outgoings.*

The entry for B14 is:

$\text{=B4} - \text{B13}$

and the block entry for the rest of the row (starting at C14) is:

$\text{=B14} - \text{C13} + \text{C4}$

The sheet shows some interesting projections:

- The maximum overdraft (about £2800) will arise in April 1994. Myra will be wise to get the bank to agree to an overdraft limit considerably higher (to allow for unforeseen events).
- If all goes to plan, the overdraft will be paid off by June 1995, Of course the whole sheet carries no guarantee of accuracy. All it shows is the eventual outcome if Myra's estimates turn out to be correct.

The sheet is useful for 'what if' questions. Some possibilities Myra might like to investigate are:

- raising the fee to £5.25.
- closing down in the winter months, when custom is poor. This will not save any rent, but will allow Myra to economise on heating and running expenses. It will also let her take another job.
- putting up a large neon sign to advertise her presence. This will cost maybe £1000, but will increase the numbers of clients by some 15-20%.

The overall result of the financial forecast fills Myra with confidence. Some people believe spreadsheets are even better than crystal balls at telling the future.

Summary

In this example we have looked at a simple example of financial forecasting, which is perhaps the most common application of a spreadsheet. Remember that the results are only as good as your estimates of the data.

This set of ten examples has given you a good start in the use of spreadsheets; but there are some topics we have barely mentioned. These topics are described in the second part of the tutorial which you should browse through when you have the time. Many useful techniques are described.

		B	C	D	E	F	G	H	I	J	K	L	M
1	Month	Aug '93	Sept '93	Oct '93	Nov '93	Dec'93	Jan '94	Feb '94	Mar '94	Apr '94	May '94	June '94	July '94
2	Est. no. of clients	90	180	120	80	40	20	20	30	120	280	240	270
3	Unit charge	£5.00											
4	Total income	£450.00	£900.00	£600.00	£400.00	£200.00	£100.00	£100.00	£150.00	£600.00	£1400.00	£1200.00	£1350.00
5	Outgoings:												
6	Rent	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00
7	Key money	£500.00											
8	Intial expenses	£500.00											
9	Monthly expenses	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00
10	Heating	£25.00	£25.00	£50.00	£50.00	£50.00	£50.00	£50.00	£50.00	£25.00	£25.00	£25.00	£25.00
11	Pay	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00
12	Bank interest		£22.00	£15.44	£15.25	£19.05	£26.93	£36.97	£47.21	£56.66	£56.79	£40.93	£28.74
13	Total outgoings	£1550.00	£572.00	£590.44	£590.25	£594.05	£601.93	£611.97	£622.21	£606.66	£606.79	£590.93	£578.74
14	Bank balance	£(1100.00)	£(772.00)	£(762.44)	£(952.69)	£(1346.74)	£(1848.68)	£(2360.65)	£(2832.86)	£(2839.52)	£(2046.31)	£(1437.24)	£(665.98)

		N	O	P	Q	R	S	T	U	V	W	X	Y
1	Month	Aug '94	Sept '94	Oct '94	Nov '94	Dec '94	Jan '95	Feb '95	Mar '95	Apr '95	May '95	June '95	July '95
2	Est. no. of clients	270	240	150	60	60	30	30	40	180	240	300	300
3	Unit charge												
4	Total income	£1350.00	£1200.00	£750.00	£300.00	£300.00	£150.00	£150.00	£200.00	£900.00	£1200.00	£1500.00	£1500.00
5	Outgoings:												
6	Rent	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00
7	Key money												
8	Intial expenses												
9	Monthly expenses	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00	£25.00
10	Heating	£25.00	£25.00	£50.00	£50.00	£50.00	£50.00	£50.00	£50.00	£25.00	£25.00	£25.00	£25.00
11	Pay	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00	£250.00
12	Bank interest	£13.32	£0.00	£0.00	£0.00	£0.00	£0.00	£0.59	£9.10	£16.78	£10.12	£0.00	£0.00
13	Total outgoings	£563.32	£550.00	£575.00	£575.00	£575.00	£575.00	£575.59	£584.10	£566.78	£560.12	£550.00	£550.00
14	Bank balance	£120.70	£770.70	£945.70	£670.70	£395.70	£(29.30)	£(454.89)	£(838.99)	£(505.77)	£134.12	£1084.12	£2034.12

Figure 10.1

Techniques for Schema2

Part 1 of this manual used a sequence of linked examples to demonstrate many of the basic techniques you need for Schema2. In Part 2, we discuss some of the advanced techniques which make Schema2 more powerful and easier to use.

Part 2 is written in a more formal style than the examples in Part 1. Instead of giving you just an overview of any topic, we say everything about it that needs to be said.

We leave the College of Irrational Sciences, where they are content with the simple techniques that will support most everyday applications. From this point onwards, our examples will be appropriate to the subject being discussed, but may not be related to each other or to anything else.

If you have read the examples module carefully, you already know enough about Schema2 to make good use of it in many practical applications. However, further study will quickly repay your effort by making your work faster, and more flexible. Your printed reports will gain a fully professional appearance.

We suggest you start by scanning quickly through the various sections to get an overall view of Schema2's advanced features. Then pick areas which you want to study in detail. As you read, try out the techniques on Schema2 with practical examples. This will nearly always clear up any problem you may have with understanding.

Between them, Parts 1 and 2 of the tutorial cover every aspect of Schema2 with two exceptions:

- Standard functions such as **SIN** or **PMT** are not described in detail. Full particulars are given in Appendix A.
- Macros are an extremely important part of the Schema2 system. They give it the full power of a standard programming language and make it possible to build a wide range of computer applications.

However, the writing of macros does require some programming expertise. Many people will not want to write macros or need their power, so we have decided not to include details about macros in the standard package, thus making a small contribution to the environment and rain forests.

Macros are fully discussed in the Macro Manual, which is published separately. Registered Schema2 owners can obtain copies free of charge from:

Techniques 1 - Selecting blocks of cells

You must select a rectangular block of one or more cells before you can enter a formula, print or draw a graph. You can only select one block at any time.

To select a block of cells use one of the following methods:

Selecting a single cell

Click **SELECT** on the cell.

Selecting a complete row

Click **SELECT** on its row label (in the margin).

Selecting a complete column

Click **SELECT** on its column heading (in the margin).

Selecting a block by dragging

Press **SELECT** on a cell at one corner of the block and drag the mouse pointer to the cell at the opposite corner. The corner of the selected block follows the mouse. If you move the mouse to the edge of the window, the spreadsheet scrolls automatically.

Changing a selected block with the Adjust button

Move the mouse pointer to a cell and click **ADJUST**. The selected block changes to include this cell as one of its corners.

Selecting a set of columns by dragging

Press **SELECT** on a column margin heading and drag the mouse pointer left or right to a different margin heading. This selects a block with all the columns you have dragged over.

Selecting a set of rows by dragging

Press **SELECT** on a row margin heading and drag the mouse pointer up or down to a different margin heading. This selects a block with all the rows you have dragged over.

Selecting a set of columns with the Adjust button

Select a complete column, move the mouse to a different column heading and click **ADJUST**.

Selecting a set of rows with the Adjust button

Select a complete row, move the mouse to a different row heading and click **ADJUST**.

Selecting all the cells in the sheet using the menu

Choose the Edit -> Select all entry in the main menu. The keyboard short cut for this action is **Ctrl-A**.

Selecting all the cells in the sheet using the mouse

Click **SELECT** in the **neutral** area (the small rectangle above the row labels and to the left of the column labels).

Combined block selection moves

Large blocks can usually be selected by a combination of simple moves, without having to move over the whole sheet.

Example - To select all the cells in the top row except A1:

1. Click **SELECT** in the left-hand margin to select the whole of the top row.
2. Click **ADJUST** on B1.

Example - To select all the cells from B2 to the bottom right cell of the spreadsheet:

1. Click **SELECT** in the neutral area.
2. Click **ADJUST** on B2.

Techniques 2 - Using the mouse to enter formulae

You can use the mouse instead of the keyboard to enter functions, macros, cell names and ranges. On the other hand you must always type numbers, text and most mathematical signs as you cannot enter them with the mouse.



Entering a function with the mouse

You can find the names of all the standard functions in a menu tree. Click **SELECT** on the **Function icon**, marked **f()** in the input area (see Figure 2.1). This action displays a pop-up menu with a list of different function groups such as Trigonometry, Array and so on. There are two additional groups with the names Macros and Sheet variable; we will say more about these groups later.

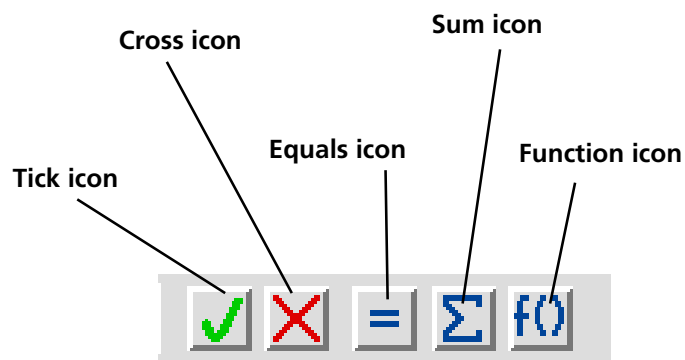


Figure 2.1

Choose a group, move across to the submenu, and choose the function you want to use.

The name of the function appears in the input area, followed by brackets for its argument or arguments. If the function expects more than one argument, the separating commas are loaded automatically. You just need to fill in the arguments themselves.

Example - To fill a cell with the formula =MID(D5,4,7):

1. Click on the Function icon. This displays the pop-up menu shown in Figure 2.2 on the right.
2. Display the submenu from String. It is shown in Figure 2.3.
3. Choose MID. The input area is loaded with the sequence:

=MID(, ,)

and the cursor is placed between the left-bracket and the first comma.

4. Enter the cell name D5 (or click **SELECT** on D5).
5. Skip the comma using the right cursor key or the mouse.
6. Type the number 4.

Figure 2.2

7. Skip the comma using the right cursor key or the mouse.
8. Type the number 7.
9. Press **Return** or **Tab**, or click **SELECT** on the tick icon.

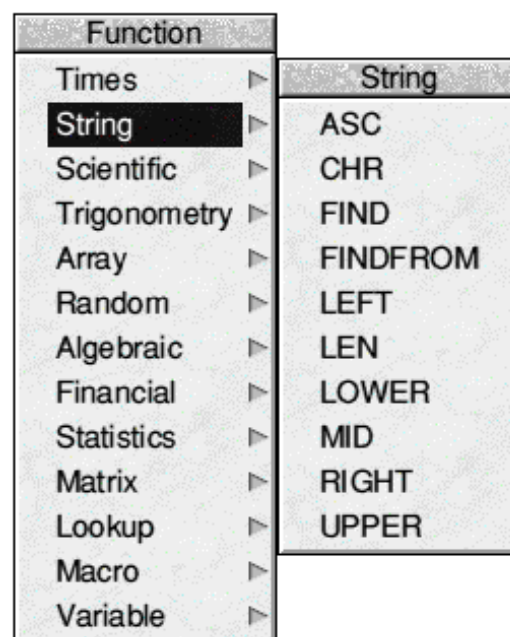


Figure 2.2



Using the Sum icon

Schema2 provides a special icon for this function. Click **SELECT** on the Sum icon to place the sequence:

=SUM()

into the input area.

This has the same effect as selecting **Function -> Array -> SUM**.

Entering a macro into a formula

Click **SELECT** on the Function icon; the menu which appears contains the entry Macros, and its submenu holds the names of any available macros. If no macros have been loaded the Macros entry is greyed out.

Macros perform a similar role to functions but are not a fixed part of Schema2. To find out more about this, see the reference section on macros.

Entering a sheet variable into a formula

If you have defined any sheet variables (see Section 19) then the pop-up menu which appears when you click on the Function icon contains the entry Variable. This leads to a submenu which lists all the sheet variables on this spreadsheet. Choose one of the names to have it included in the formula.



Entering cell and block names

To use the mouse for entering cell and block names the system must be in *mouse mode*, which is enabled when the Equals icon changes to an arrow pointer.



As soon as you start typing a formula (or if you click on the Equals icon), Mouse mode is enabled. Click **SELECT** on a cell; it is not selected but its name is entered into the input area as if you had typed the name.

Schema2 puts an = sign or a + sign in front of a cell name whenever it is logical to do so.

Example - To enter the formula =A1+A2:

1. Click **SELECT** on the Equals icon.
2. Click **SELECT** on cell A1.
3. Click **SELECT** on A2.

To enter a mathematical symbol other than '+' before a cell name, type the symbol and then click on the cell.

Example - To enter the formula =A1-A2:

1. Click **SELECT** on the Equals icon.
2. Click **SELECT** on A1.
3. Type a minus sign.
4. Click **SELECT** on A2.

To enter a range using the mouse, place the pointer on one corner of the range and drag it to the opposite corner. The cells in the range turn grey to show that they are being entered.

Example - To enter the range A1..D5:

1. Press **SELECT** on cell A1.
2. Drag to D5 and release the mouse button.



Terminating a formula with the Tick icon

Click **SELECT** on the Tick icon. The Tick icon is similar to the **Return** or the **Tab** key: it enters the data on the input line into the main body of the spreadsheet. However, unlike these two keys, the Tick icon does not move the selected cell.

As a side effect, the Tick icon returns the system to normal, or non-mouse mode.



Cancelling a formula with the Cross icon

Click **SELECT** on the **Cross icon**.

The Cross icon removes all existing data on the input line, leaving it empty. The system returns to non-mouse mode, but the current cell selection is not changed.

Techniques 3 - Changing Column and Row Dimensions

This section describes how to change the size of the cell grid in any sheet. The creation of big cells, which occupy several adjacent positions in the grid, is discussed in Section 6.

Choosing measurement units with the Choices Dialogue

You measure spreadsheet row and column dimensions in **inches**, **centimetres**, millimetres or points. (A point is 1/72 part of an inch). You can choose the units from the Choices Dialogue, which is described in Section 26.

The rest of this chapter assumes that you have selected inches as the unit of measurement, and quotes all sizes in inches.

When you create a new sheet, all the columns are a standard width determined by the **Default Document** - usually 0.8 inches. The usual default setting for the row height is 0.29 inches.

Columns can be of any width down to 0.25 inches, and up to a size which still allows the sheet to be drawn on the screen. The rule is that the left-hand sheet margin, including any columns which have been 'locked', must be completely visible within the width of the screen.

Rows must be more than 0.2 inches high, but may not be so high that they prevent the sheet from being drawn in the standard manner ie larger than the screen.

This Section describes the techniques you need to alter column dimensions. Row dimensions can be changed in the same way - just read 'rows' for 'columns' and 'height' for 'width'.

There are several ways you can change the width of a column or a group of columns:

Setting a new width by menu

Select the column or columns you want to change, and then choose the writable menu item **Edit -> Columns -> Width**.

The menu item gives the current width of the selected column in inches (or the chosen unit of measurement). If more than one column is selected, the number you see is the average width of the selected columns.

Change the width to the value you need, and press **Return**. The selected column(s) will be redrawn to the new width.

Changing the width of a column by dragging

Put the mouse pointer near the right hand edge of the column name in the top margin. The pointer changes shape to a double-headed horizontal arrow, as shown in Figure 3.1. Now press **SELECT** and drag the cell boundary to any position you want. When you release the button, the column width is adjusted to the new size.

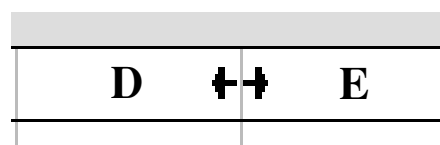


Figure 3.1

Changing the width of a group of columns by dragging

Select a group of columns, and put the mouse pointer near the edge of the column name of the right-most column. The pointer changes shape to a double-headed horizontal arrow. Now press **SELECT** and drag the column border to a new position. When you release the button, all the selected columns will change to the same width. These widths, when taken together, just fill up the space between the left of the first selected column and the new column border.

You can use this technique to make sure that all the columns in any group have the same width. Select the group of columns and drag the right-most column border in the top margin.

If the column border you drag is not at the extreme right of the selected group of columns, then the columns beyond the selected border do not change in width.

Adjusting the width of a column to its widest data item

Double-click any column name in the top-margin, and the width of that column changes so that the widest data item in the column fits comfortably.

Note: Methods which involve dragging are not suitable for enlarging columns beyond the displayed width of the sheet. In particular, the width of the last column cannot be increased by dragging. You must change its dimensions by using the menu option.

Everything we have said about changing column widths also applies to row heights.

Techniques 4 - Changing sheet layout

When you create a new sheet in the normal way by clicking on the Schema2 icon, it starts its existence as an exact copy of a special spreadsheet called the **Default Document**. Methods of changing the Default Document are described in Section 26.

There are several ways you can modify the dimensions of individual sheets without altering the default document.

Using the Create New Spreadsheet dialogue.

To use the Create New Spreadsheet dialogue bring up the icon bar menu and choose the Create option. This is shown in Figure 4.1.

Use this dialogue to create a sheet with any name and any number of rows and columns up to 30000. The other properties of the sheet (such as the size of each cell) are still determined by the Default Document.

The name of the sheet is limited to eleven characters and should not include any spaces or points.

Figure 4.1

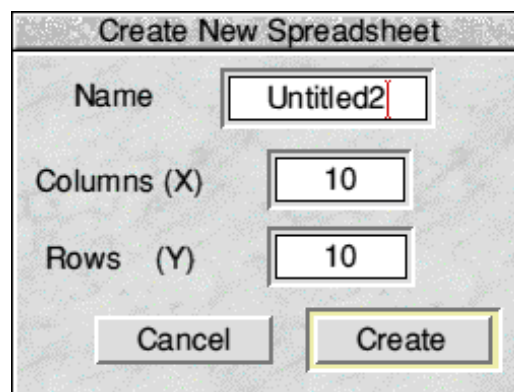


Figure 4.1

The total number of cells in the sheet is limited by the amount of memory available. If you set one dimension to 30000, the other one may well be limited to a small number such as 1 or 2.

Inserting rows

To insert one or more new rows anywhere in the sheet:

1. Select the row immediately above the place where the new rows are to be added.
2. Open the main menu and choose the writable icon from **Edit -> Rows -> Add rows.**
3. Change the contents of the icon to the number of rows to be added.
4. Click the mouse or press **Return**.

When you add rows, all the material below the selected row is moved down by the appropriate number of cells. References are adjusted accordingly.

If a block reference spans the place where a row is inserted, it is adjusted automatically.

Example - Cell B20 contains the formula

`=SUM(B1..B19)`

When a new row is inserted - say - at row 15, the sum formula is moved to cell B21 and adjusted to read

`=SUM(B1..B20)`

Deleting Rows

To delete one or more new rows anywhere in the sheet:

1. Select the row(s) you want to delete.
2. Choose the main menu option **Edit -> Rows -> Delete rows.**

When you delete rows, all the cells below are shuffled up, and the references are adjusted automatically.

Inserting columns

To insert one or more new columns anywhere in the sheet:

1. Select the column immediately to the left of the place where the new columns are to be added.
2. Open the main menu; choose the writable icon at **Edit -> Columns -> Add columns.**
3. Change the contents to the number of columns you want to insert.
4. Click the mouse or press **Return**.

When you add columns, all the material to the right of the selected column is moved towards the right by the appropriate number of cells. References are adjusted accordingly.

If a block reference spans the place where a column is inserted, it is adjusted automatically.

Deleting Columns

1. Select the column(s) to be deleted.
2. Choose the main menu option **Edit -> Rows -> Delete Columns.**

When you delete columns, all the cells beyond are shuffled to the left, and the references are adjusted automatically.

Techniques 5 - Changing the margin content

This explains how you can alter the contents of the margins of a spreadsheet. You can lock rows and columns, and hide the standard row and column labels.

Hiding and restoring System labels

To hide or restore the system labels - **A, B, C...** in the top margin, and **1, 2, 3...** in the side margin:

1. Choose the menu option **Utilities -> Labels**. Alternatively, use the keyboard short cut, which is **Ctrl-O**.

This menu item is a toggle so a second click brings the labels back.

If you remove the system labels and do not lock any rows or columns, the neutral area vanishes and cannot be used to select all the cells. However, you can still select all the cells by choosing the main menu entry **Edit -> Select all** (the keyboard short cut is **Ctrl-A**).

Locking rows or columns

This tells you how to lock and unlock rows. To lock columns use the same method, just read 'columns' for 'rows' and 'side margin' for 'top margin'.

If you lock a row of cells, Schema2 keeps a copy of the row in the top margin. The copy is always visible even if the original row, in the body of the sheet, is scrolled out of sight. Locked rows serve as user-defined column names.

To lock a row or two adjacent rows:

1. Select the row(s) to be locked.
2. Choose the menu item **Edit -> Rows -> Lock rows**.

To lock a column, or two adjacent columns:

1. Select the column(s) to be locked.
2. Choose the menu item **Edit -> Columns -> Lock columns**.

You cannot lock a group of rows if their combined height is greater than the screen height. A similar rule applies to columns.

To unlock a row or group of rows:

1. Make any selection which does not include a complete row.
2. Choose the menu item **Edit -> Rows -> Free rows**.

To unlock a column or group of columns:

1. Make any selection which does not include a complete column.
2. Choose menu item **Edit -> Columns -> Free columns**.

The Appearance of Row and Column Labels

The row labels **1 2 3...** and the column labels **A B C...** are normally displayed in Labels style (See Section 10 for a discussion of styles). Any change to the definition of Labels will affect the appearance of the row and column labels.

If no Labels style is defined then Plain style is used as a default for the labels.

Techniques 6 - Views and Spreadsheets

Views

When you handle large spreadsheets, it is often useful to be able to see more than one area of the sheet at the same time.

A view of a spreadsheet is a window in which part of the spreadsheet is shown. A spreadsheet can have up to four views on the screen at the same time. The views can be scrolled and sized independently. However they all refer to the same spreadsheet, so any change to it will be reflected at the same time in all the views.

Making a new view

To make a new view of a spreadsheet, either:

- Bring up the New view menu from the bar icon menu, and choose the name of the spreadsheet.

or

- If there is already a view open for the spreadsheet, click **MENU** on it and choose **View -> New view** menu from the main menu.

If a sheet has more than one view open, the title bar of each view shows the number of views on to that sheet; but the views themselves are anonymous; they are **not** numbered 1,2,3... Each view supports all the operations you can do on a spreadsheet.

Closing a view

To close a view, click on the close icon at the top left of the window. If the Discard sheet on close option has been set in the Choices dialogue, and this is the only open view on the spreadsheet, then Schema2 **discards** the spreadsheet, after giving a warning if the sheet is unsaved. Otherwise the view is closed and the spreadsheet is left unchanged.

It is worth noting that views and sheets are distinct entities. If the Discard sheet on close option has **not** been set in the Choices dialogue you can discard all the views of a sheet, but the sheet itself still remains in the system and you can bring back a view with New view. This lets you keep a large number of linked spreadsheets in memory without crowding the screen with windows.

Discarding Spreadsheets

When a spreadsheet is **discarded** Schema2 removes the spreadsheet data from the random access memory of the machine. This frees this memory for use by other spreadsheets or other applications. Before you discard a spreadsheet you would normally save it to a disc or other type of storage medium. When you discard a spreadsheet any open views are automatically closed.

To discard a spreadsheet, do one of the following:

- Choose the spreadsheet from the Discard submenu of the icon bar menu.

or

- If there are open views of the spreadsheet click **MENU** on one of them and choose **File -> Discard sheet** from the main menu.

or

- If there is only one view open of the spreadsheet and Discard sheet on close has been set in the Choices dialogue, close the view by clicking on the close icon of the window.

If Schema2 finds that the sheet to be discarded has been changed (and not saved) it asks you to confirm that you really want the new version of the sheet to be destroyed.

Appearance of views

There are a number of ways in which you can modify the appearance of views from the View submenu. These modifications apply to all views simultaneously and most of them affect **printing** and **Drawfile** export.

Scale view

You can choose the scale that the spreadsheet is displayed on the screen to be anything between 25% and 200%. This does not affect the scale at which the sheet is printed.

Grid

You can switch grid dividing cells on and off using **View -> Grid** or key **F10**.

Labels

You can switch the cell labels on and off using **View -> Labels** or **Ctrl-O**. For more details see Section 5.

Icon ribbon

By default the icon ribbon is displayed in the entry area. You can turn the icon ribbon off by choosing **View -> Icon ribbon** from the main menu. This lets you see a little more of your spreadsheet on screen at any one time, but you can't use the icon ribbon. To display the ribbon again choose **View -> Icon ribbon** from the main menu again.

The icon ribbon does not appear when the spreadsheet is printed or exported as a Drawfile.

See Section 27 for details about modifying the icon ribbon.

Formulae

Sometimes you need to see the formulae used to create a sheet rather than the values. To change the display to formulae choose **View -> Formulae** from the main menu. To change back to normal values display **View -> Formulae** from the main menu again.

You can also obtain a record of the formulae in your spreadsheet by using **File -> Export -> Formulae**.

Techniques 7 - Creating and Destroying Big Cells

A **big cell** is one which occupies several adjacent grid positions. Big cells are rectangular, and the home position of a big cell is always one of the grid locations within its boundaries. The cell is usually known by its home position.

A big cell hides other cells. For example, a big cell might be rooted in B4, and extend to use the sheet area normally allocated to C4, D4, B5, C5 and D5.

Big cells are used to hold large items such as long strings, characters in large typefaces, graphs, or multi-line text.

There are two types of big cells: **automatic** and **explicit**.

Creation of Automatic big cells

Whenever a cell is filled with a string item which is too long to fit, the system tries to turn the cell into a big cell of sufficient size. Starting with the original cell as a root location, the cell tries to grow over neighbouring empty cells until it is large enough to display the whole item. Growth is always upwards for large characters, and can be towards the right, the left, or both, depending on how the string item is aligned (items which are left-aligned grow towards the right).

A big cell can never grow over another cell which has visible contents. If a non-empty cell is found during expansion, growth stops.

Cells which contain numerical items never grow automatically, even if the normal cell is too small to hold the contents. Expansion is only used for cells which hold strings. In this context, remember that a temporal value (date) is treated as a **numerical** item.

Once a big cell has been created you can't directly select any of the invisible cells behind - if you click in the place you expect the cell to be, you select the big cell instead. However, most other selection mechanisms, such as row or column selection, **Adjust** click or dragging can still select invisible cells.

The extent of an automatic big cell can be affected by three types of operation:

- adjusting its contents or style.
- placing a visible item in any of the invisible cells behind the big cell.
- changing a row or column size.

Creating and destroying Explicit big cells

You can create a big cell with a size which does not depend on its contents. This is particularly useful for blocks of text or embedded graphs.

To create a big cell:

1. Select an area which contains only empty cells, or at most **one** filled cell.
2. Choose the **Edit -> Big cell** entry in the main menu or press **Ctrl-F**.

The cell boundaries within the selected area vanish, clearly labelling the area as a big cell.

If the original area was empty, the big cell is rooted in the top left hand corner of the selected area.

If the area contained a single filled cell, the big cell will be rooted on that cell, and keep the same contents.

To destroy a big cell, select it and choose the **Edit -> Big cell** entry.

Multi-line text

If you place a long string into a big cell, it is split into several lines of text, ensuring that individual words remain intact if at all possible. The system interprets the sequences **IM** and **Im** as forced newlines, allowing the material to be divided into paragraphs.

A long string in a big cell allows for four types of horizontal alignment:

- to the left
- to the right
- centred
- justified to both margins

Similarly, a long string can be vertically aligned in three ways:

- top-alignment: the first line is lined up to the top edge of the cell
- middle-alignment: the string is centred (vertically) in the cell
- bottom-alignment: the last line of the string is lined up to the lower border of the cell

Planting graphs and pictures into cells is discussed in Section 21.

Explicit big cells do not change their size, unless you alter the column width or row height of any of the underlying columns or rows. They can be used to plan layout.

The two types of big cell are illustrated in Figure 7.1:

	A	B	C
1	This is an automatic big cell		
2			
3	This is a big cell which has been set up explicitly		
4			
5			

Figure 7.1

The big cell mechanism is designed to handle items of **exceptional** size, such as titles which spill over several rows or columns. There is no limit to the number of big cells in any sheet, but their overall effect is to consume memory and slow down the program. If you expect your sheet to hold many cells with large data items, you should increase the basic cell size by changing the column and row dimensions.

Techniques 8 - Number format dialogue

In Schema2 you can set the format of numbers with the Number format dialogue. This dialogue is part of the Effect submenu.

Setting a Number format

1. Select a group of cells in which you want to set the number format.
2. Click **MENU** on the sheet and choose **Effect -> Number format** or press **F8** or click **SELECT** on the Number format icon in the icon ribbon.
3. Set up the required format using the dialogue (Figure 8.1).
4. Click on the **Apply** button.

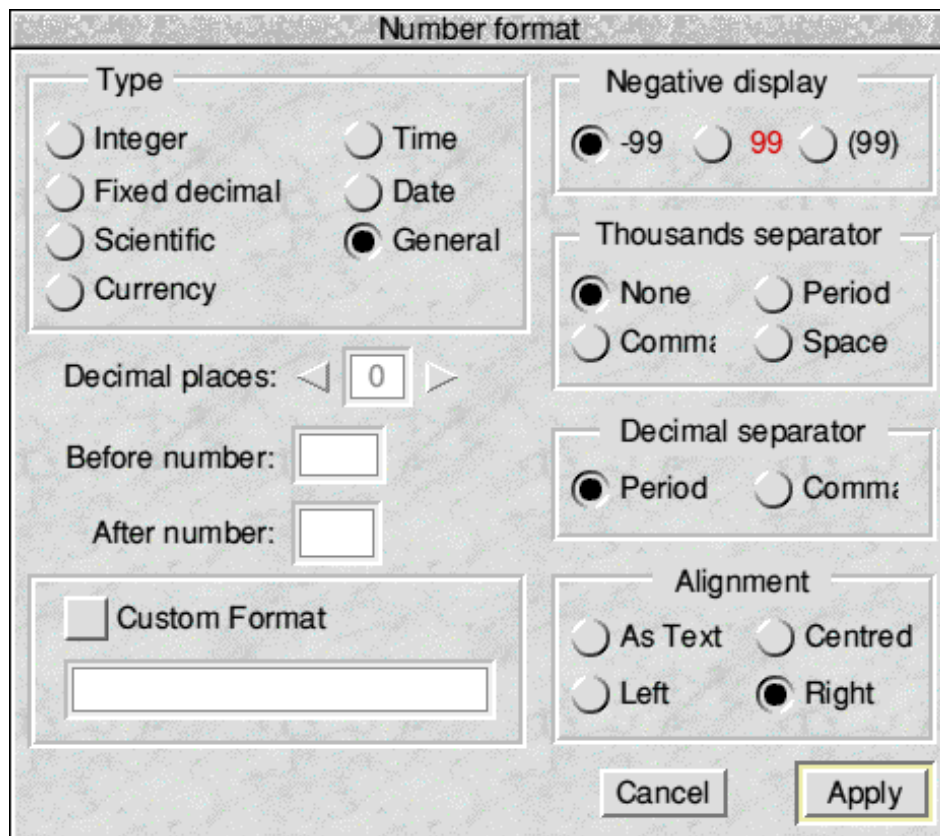


Figure 8.1

The Number format dialogue is divided into several parts, including a number of standard options and a custom format field. In most cases, you can set a number format as a combination of standard options, but in special cases you can format numbers with even greater flexibility using a Custom format string. Custom formats are discussed in Section 9.

The standard options in the Number format dialogue are as follows:

Type

This group consists of a set of radio buttons - that is, only one of them can be selected at any time.

The group determines the type of conversion. Possible types (with examples) are shown below:

Type	Example
• Integer	345
• Fixed decimal	67.6500
• Scientific	6.77 e+2
• Currency	\$56.34
• Time	4.05pm
• Date	27th August 1993
• General	67.65

Note the following points:

Any value displayed as an Integer is rounded to the nearest whole number.

A Fixed decimal number is given exactly the number of decimal places given in the Decimal places field. The value is rounded so that the error in the last place is at most half a unit.

Similarly, a Scientific number is displayed with the number of decimal places given in the Decimal places field. The scientific format is specially useful for very small or very large quantities.

A value expressed as a Currency uses the stated number of decimal places, and allows for short arbitrary strings (such as £, HK\$ or DM) to be placed before or after the value (or both).

A value displayed as a Time should be a **temporal value**. The display shows hours, minutes and seconds in the 24-hour clock. The order is determined by the environmental variable **Schema2\$TimeFormat**.

A value displayed as a Date should also be a temporal value. The display shows a day number, a month and a year, in the order specified by the environmental variable **Schema2\$DateFormat**. Note that it will not often be necessary to change this variable or **Schema2\$TimeFormat** since alternative time and date formats can be set with Custom format strings.

A General value uses just as many decimal places as it needs, suppressing trailing zeros. There is a limitation to about 15 significant figures.

General is the default format, used for cells to which no other number format has been assigned. This is true unless you alter the definition of Plain style - see Section 10 on styles.

Decimal Places

This writable field selects the number of decimal places used when a number is displayed in any of the following types:

- fixed decimal
- scientific
- currency

You can alter the number in the field by clicking on the arrow buttons at the sides.

Before number

Use this field to specify a short string (up to 3 characters) which is displayed *before* a number. The field may be used for currency designations or physical descriptions, such as £ or pH.

After number

This field lets you specify a short string (up to 3 characters) which is displayed *after* a number. This is often used for physical units or currency designations such as Kg. or FF.

Negative display

This field contains radio buttons which let you choose three different ways of representing negative numbers:

- prefixed by a minus (-) sign
- in red
- in brackets

The field has no effect on the way that zero or positive numbers are displayed.

Thousands separator

This field has four radio buttons and gives you a choice of characters to separate groups of three digits in a number. The following are available:

- none (as in 12345.678)
- comma (as in 12,345.678)
- period (as in 12.345,678)
- space (as in 12 345.678)

Decimal Separator

This field has two radio buttons and lets you choose a Period (dot) or a Comma for the decimal point. Schema2 will not let you use the same character as you have specified for the thousands separator.

Alignment

This field consists of four radio buttons, and allows you to override the alignment set by the general effects system.

The alignment set by the format dialogue only applies to numbers. This makes it possible for the same cell to use left-alignment for strings but right-alignment for numbers. The following possibilities are available:

- use existing alignment as set for text
- align left
- align right
- centre

This feature lets one set of effects or style display cells with text aligned to the left and numbers to the right. This is the default for Plain style.

Techniques 9 - Custom formats

A Custom format is a string of characters which forms part of the effect attached to any cell. Custom formats control the way that numbers and temporal values are displayed, and allow a wider choice than you can get with the options on the Number format dialogue.

Setting a Custom format

To set a Custom format:

1. Select a cell or group of cells to which the Custom format is to apply.
2. Bring up the Number format dialogue (see Section 8).
3. Click the Custom format button.
4. Enter the Custom format in the writable icon.
5. Click on the **Apply** button.

The meaning of a Custom format

You build a Custom format from two kinds of element: dumb characters and commands.

A **command** consists of a % sign, usually followed by a group of one or more characters. A **dumb character** is any symbol which is not part of a command.

Example - Consider the sequence **Score=%0.0f%% (PASS)**

Here the dumb characters are **S c o r e = (P A S S)** and the commands are **%0.0f** and **%%**

This Custom format string displays a number in the format

Score=74% (PASS)

When a Custom format string is used to display a number, all the dumb characters are simply copied.

Displaying a percent sign

Since the % sign is used to start commands, it cannot be used directly as a dumb character. The command **%%** displays a single % sign.

Alternative for zero

When a number is exactly zero, you may need to show it in some way which is different from a simple 0. Use the command **%no**, followed by the alternative representation. Terminate the form with a comma.

Example The command **%noNIL,** displays zero as **NIL**

Layout of numbers

There is a small group of commands which indicates how to format the number being displayed. They are:

%0.0f	Display as an integer as in 1234
%0.<n>f	Display as a decimal number with <n> decimal places. <n> stands for any integer. For example %0.7f might produce 3.1415927
%0.<n>e	Display as a number in scientific format with <n> decimal places. <n> is any integer. For example %0.4e might produce 5.1235+e9
%ge	Display a number in general form, with as many decimal places as it needs.

If a Custom format string holds more than one number layout command, the number will be displayed once for each command. For example, the string:

%0.0f (alias) %0.4f could display **3(alias)3.1416**

Layout of dates and times

There is a group of commands which interpret the number as a temporal value and extract one or other component of the date or time. See Section 17 for a description of temporal values.

Example - The value of **NOW()** as I write this is **63935790224**

The command **%ce** if applied to this value, produces the century number: **19**. Similarly, the command **%yr** applied to the same number, generates the year number within the century: **93**

Other commands extract the month, day, hour, minute and second. Yet others produce the name of the month, the month number in Roman characters, or the name of the weekday. The full list is:

%ce	Century number
%yr	Year number in century (00 to 99)
%mn	Month number in year (01 to 12)
%mo	Month name (in English)
%m3	First three letters of month name
%ro	Month number in Roman numerals (I to XII)
%dy	Day of the month (01 to 31)
%dn	Day number of the year (001 to 366)
%st	English suffix for day of the month: st, nd, rd, th.
%we	Day of the week, in English
%wk	Week number of the year (01 to 52)
%w3	First three letters of the weekday
%wn	Weekday as a number (1 = Monday - 7 = Sunday)
%24	Hours in a day (00 to 23)
%12	Hours in a half-day (01 to 12)
%am or %pm	One of am or pm
%mi	Minute of the hour (00 to 59)
%se	Second of the minute (00 to 59)

You can string these commands together, with suitable punctuation, to show a date or time in any imaginable format.

Example - **%24%mi** displays the time in 24 hour notation, as in **0810**

In their two-letter forms, some of these commands always produce two (or three) figure numbers in which the first figure may be zero. For example, the year number of **2001** will be **01**.

With these commands, a **z** just after the **%** sign suppresses the initial zero(s). This applies to **%ce**, **%yr**, **%mn**, **%dy**, **%wk**, **%dn**, **%24**, **%12**, **%mi** and **%se**.

More examples of Custom format strings, and their possible results:

%12 o'clock	will display	08 o'clock
%z12 o'clock	will display	8 o'clock
%we, the %zdy%st day of %mo in the year %ce%yr	results in	

Saturday, the 28th day of August in the year 1993

Note: When interpreting a temporal value Schema 2 takes into account the **time mode** for that spreadsheet. Usually this is hidden from the user as the **Time functions (NOW()** etc.) return numbers which take this into account.

Colour of Negative numbers

You can arrange for all negative numbers to appear in a special colour, such as red. Use the command **%cn** followed by an 8 digit hexadecimal number, as in

%cn00ffff00

Each pair of digits in the number has a special meaning:

- The first pair gives the intensity of **blue**, on a scale between **0** and **ff** (maximum).
- The second pair gives the intensity of **green**, on a scale between **0** and **ff**.
- The third pair gives the intensity of **red**, on a scale between **0** and **ff**.
- The fourth pair is always **00**.

Our example gives the following intensities:

blue = 0%, green = 100%, red = 100%

This specifies **yellow**, which is a mixture of red and green.

Some useful colours and their codes are:

Black	00000000
Red	0000FF00
Green	00FF0000
Yellow	00FFFF00
Blue	FF000000
Magenta	FF00FF00
Cyan	FFFF0000
White	FFFFFF00

You can get a wider range of colours and shades by using intensities other than 100%. Here is a list of hex numbers corresponding to various intensities:

Intensity(%)	0 10 20 25 30 33 40 50 60 67 70 75 80 90 100
Number (hex)	00 17 33 40 4b 55 66 80 97 aa b5 c0 cd e9 ff

Example - *Pink is made up of 100% red, 75% green and 75% blue. To have negative numbers shown in pink, use the command:*

%cncocoff

The colour you choose for a negative number overrides the colour specified for the text in the cell. There are two points you should bear in mind:

- When a negative number is displayed in a special colour, the - sign is omitted.
- If the special colour is the same as the background colour of the cell, the value is invisible. Beware!

Extra characters for negative values

Another way of displaying negative numbers is to add special characters (such as brackets) either before or after the value itself. Use the command **%br** followed by a string to be placed before the number, and a second string to be placed after it. Both strings end with commas, and may contain no commas.

Examples	%br(,),	will display (9) for -9
	%br,(Overdrawn),	will display £56.78(Overdrawn) for -£56.78

The **%br** command does not affect positive numbers or zero.

Thousands separator

This command determines the character which separates groups of three digits. It consists of the sequence **%ts** followed by a single character.

Example **%ts/** shows 123456789 as 123/456/789

Fraction separator

The fraction separator command fixes the appearance of the decimal point. Use **%fs** followed by a single character.

Example **%fs*** displays **34.678** as **34*768**

The default fraction separator is the normal point.

Alignment

The alignment commands show how numbers should be horizontally aligned in their cells. These commands override the alignment effect attached to the cell as a whole. They are:

%a0	align to the left
%a1	align right
%a2	align centrally

Sequence

The custom format commands should be entered in the order that you expect them to appear in the cells eg **%ts,%fs#%ge%a1**

If you get a blank cell then you have forgotten to include **%ge** or an equivalent such as **%0.3**

Techniques 10 - Styles

When you build a large spreadsheet you may notice that you keep having to apply the same sets of effects to many different cells. Schema2 has a feature which can automate this work for you. This feature is **styles**.

A style is a named collection of effects. A large sheet might use several different styles. For example, there may be one style for headings (large characters), one for sub-headings (bold), one for currency (£ sign and two decimal places) and one for simple numbers.

When you create a new sheet you can define the styles you need. As you enter each cell or block you can assign a suitable style to it by selecting from a menu

Later, you may decide that your headings should use a different font. If you change the style definition, all the headings using that style will change automatically - there is no need to edit each one individually.

Another important use of styles is to produce spreadsheets to a house style. You might decide that all the spreadsheets produced by your organisation should have a uniform appearance. To do this you can create your own set of styles for everyone to use. Then someone constructing a spreadsheet only needs to select the right style for each cell or area - they do not have to remember details such as the exact point size of a sub-heading.

Schema2 is very flexible in using styles and effects. You can use just effects, just styles or styles and effects according to your situation and preferences.

A style controls the following aspects of cell display:

- font name
- boldness
- italicisation
- font size
- foreground colour
- background colour
- alignment
- vertical alignment
- number format

Every cell in a sheet has an associated style and may also have one or more effects which override parts of its style. When a new sheet is opened every cell is given a default style called **Plain**. Its properties, unless you change them are:

font name	Trinity
bold	off
italic	off
font size	12 point
foreground colour	black
background colour	white
alignment	left
number format	general, numbers aligned right
graphics display	fit to cell, lock aspect ratio

This means that when you display or print a sheet without changing any styles or effects all the contents are displayed in Plain style, in Trinity 12-point using black on white. Numbers are aligned to the right, and strings to the left.

When you apply an individual effect to a cell or to a group of cells, that effect overrides the corresponding component of the style for those cells only. For example, if you reset a block of cells to 18-point font size using the Effect menu, each cell is displayed in 18-point, but with all the other features still determined by the resident style.

When you apply a new **style** to a cell, then all the details of the display are changed to conform to the new style, unless any effect has been set; then that effect is retained and overrides the effect specified in the new style.

Example - Consider the following sequence:

1. You fill B2 and B3 with two strings. They both appear in **Plain** style: black letters on a white background.
2. You change the effect in cell B3 to have a red background.
3. You define a new style called **Special**, which has green characters on a white background.
4. You apply this new style to block B2..B3.

The string in B2 will now be green on white but B3 keeps its red background and displays its contents in green on red.

There are four basic operations with styles. You can:

- **apply** a style to a cell or block of cells.
- **create** a **new** style.
- **edit** an existing style.
- **delete** a style.

You can also

- **Clear all effects** from a cell or block of cells.

Applying a style

1. Select the area of cells to which the style is to be applied.
2. From the main menu choose a style name from the **Style -> Apply** submenu.

Note that if there are effects applied to the cell(s) then they override the style.

Creating a new style

1. Choose the **Style -> New entry** in the main menu which leads to this dialogue (Figure 10.1):
2. Enter the new style name in the top writable icon.
3. If the new style is to have any resemblance to an existing style, click on the menu button labelled **Copy Attributes from Style** and choose one of the styles you see. If you miss this step, the default values of all the attributes will be taken from Plain style.
4. Click on the menu button labelled **Style Attributes**. The menu which you see displayed is the same as the normal Effects menu except that there is no mention of the Cell Border or of clearing all effects. Use the various items to set up every attribute of the new style.

You may find it useful to use the **ADJUST** button when changing more than one attribute; this keeps the menus open.

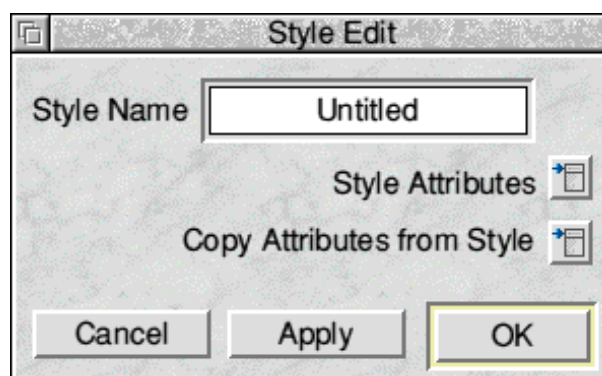


Figure 10.1

5. You now have three choices:

- OK saves the new style so that it can be used in future.
- Apply saves the style and also applies it immediately to the current selection of cells.
- Cancel discards the new style entirely.

Editing a style

You can change an existing style as follows:

1. Open the main menu and choose a style from the **Style -> Edit** menu. This brings up the Style edit dialogue with the selected style name shown in the top icon.
2. Change the name of the style (if you wish).
3. Change any attribute of the style.
4. Click on one of the **Cancel**, **Apply** or **OK** buttons.

Deleting a style

1. Choose the style from the **Style -> Delete** submenu.

You can't delete a style if there is any cell which uses it in the sheet. Plain style cannot be deleted at all.

Removing Effects

When you apply a new style to a block, you may want to remove all existing effects. To do this:

1. Choose the **Effect -> Clear all effects** from the main menu.

Setting a Labels style

Normally Schema2 displays row and column labels in Plain style. However, if you define a new style called **Labels**, that style is used for the margins instead. To illustrate this feature bring up the style dialogue, and create a style called **Labels** with the attributes of **Accountant**. As soon as you click on OK the characters in the margin change.

Sharing styles between sheets

A style is local to its own sheet, and there is no way that a cell can be set to show the style defined in a different sheet.

However, styles can easily be copied to different sheets. If you copy a block of data from sheet **A** and paste it to sheet **B**, every style in A is transferred automatically unless its name clashes with one defined in sheet B.

For example, suppose that in sheet **A** you have defined a style called **Grand**. You apply this style to a cell (say **A1**), copy the cell, and paste it into sheet **B**.

If sheet **B** does not have a style called **Grand**, this style is copied to Sheet **B** and the cell you have pasted appears in **Grand** style.

If sheet B already has a style called **Grand**, A's version of **Grand** is not copied; the pasted cell is displayed in B's version of **Grand**.

If you want to set up a style which is to be used in more than one sheet, give it a new name; don't edit one of the existing styles as it will not be transferred.

Techniques 11 - Editing tools

In this Section we discuss the editing tools you find at the top of the Edit menu. They are:

- **Cut**
- **Copy**
- **Paste**
- **Paste values** and **Paste values Transposed**
- **Move**
- **Delete**
- **Replicate**

The Clipboard

A useful feature of Schema2 is the **clipboard**. This is a hidden area of memory which can hold a copy of any block in a spreadsheet (including an entire spreadsheet - memory permitting).

Schema2 shares its clipboard with certain other applications, including **Advance**, **DataPower** and **Impression Style**. It is likely that other products will come to share the clipboard standard in the future.

The clipboard lets you move data about

- within a spreadsheet.
- between different spreadsheets.
- between different applications which conform to the clipboard standard.

The export and import facilities described in Sections 13 and 14 give you more control over moving data between Schema2 and other applications, but not so conveniently as through the clipboard.

Cut, Copy and Paste

You can Cut data from one cell or block of cells and Paste it to another position on the spreadsheet. The **cut** and **paste** metaphor comes from the printing industry where typesetters who arrange page layouts use scissors to snip out pictures or sections of text, and stick them in different places on the page.

Using cut and paste, you can change your mind about the sheet layout as often as you like. When you enter values into a cell or group of cells, and then decide that you want the data somewhere else, you can **cut** it from one position and **paste** it into its new home.

Data which you cut disappears from the sheet, but is stored in the clipboard until you are ready to put it in another position.

The Copy command is similar to cut except that it leaves the data in its original place and puts a copy into the clipboard. You can use Copy and Paste to transfer data without damage to the original.

The description of Cut, Copy and Paste we have just given must be modified slightly when the source data is a formula and not a constant value. In this case cell names in the formula may be adjusted to retain their relative positions. The exact changes made are detailed at the end of the Section.

Example - To move a block of data from A1..A7 to C3..C9:

1. Select block **A1..A7**
2. Cut
3. Select cell **C3** (or any block of cells with **C3** at the top left)
4. Paste.

The actual details of the cut, copy and paste commands are given below.



Cut

Select the block to be cut, bring up the main menu and choose **Edit -> Cut**. The information disappears from the selected block, but is stored in the clipboard. Any other data previously in the clipboard is lost.

The keyboard short cut for Cut is **Ctrl X**. Alternatively the Cut icon from the icon ribbon can be used.



Copy

Select the block to be copied, bring up the main menu and choose **Edit -> Copy**. The information is stored in the clipboard. Any other data previously in the clipboard is lost.

The keyboard short cut for Copy is **Ctrl-C**. Alternatively the copy icon from the icon ribbon can be used.



Paste

When you have cut or copied the block you want to transfer, select the top left hand cell of the destination area, bring up the main menu and choose **Edit -> Paste**. The data in the clipboard is pasted into the destination area. All previous contents of the area are destroyed. The Paste function won't let you paste a block in a position which would overhang the borders of the sheet.

The keyboard short cut for pasting is **Ctrl-V**. Alternatively the paste icon from The icon ribbon can be used.

Paste values -> Normal

Use the **Paste values -> Normal** command to paste the numbers or strings in a block rather than the formulae which calculated those contents. First, you cut or copy the contents of a block to the clipboard. Then instead of using the Paste command to paste the formula, use the **Edit -> Paste values -> Normal** command to paste the values in the cut block.

If a cell in the original block contains a simple number or string, then **Paste** has the same result as **Paste values**. However, if the original cell has a formula, the destination cell ends up with a simple value which is not recalculated when the numbers in the formula change.

Use **Paste values** to preserve copies of calculations and ensure that these results are not altered when other parts of the sheet change. This lets you run A series of *what if* experiments and keep the results of each one on the sheet.

Paste values -> Transpose

This is the same as **Paste values -> Normal** except that the block of data is transposed before it is pasted, that is rows become columns, and vice versa.

The command is found at **Edit -> Paste values -> Transpose**.

Example - You are setting up a mileage chart between various towns. In column A (starting at A2) you have already set up a list of places which starts:

Aberdeen
Birmingham
Bournemouth
Brighton
Bristol
.....

To label the chart you need the same names over the columns, starting at B1. To fill in these labels:

1. Copy column A, starting at A2.
2. Select cell B1, and choose the menu item **Edit -> Paste values -> Transpose**.

The result, with some numbers added, is shown below (Figure 11.1).

Town	Aberdeen	Birmingham	Bournemouth	Brighton	Bristol
Aberdeen	..*-	476	643	635	570
Birmingham	476	..*-	134	146	87
Bournemouth	643	134	..*-	54	107
Brighton	635	146	54	..*-	143
Bristol	570	87	107	143	..*-

Figure 11.1

Move

Use Move to shift parts of the spreadsheet around without changing the underlying logic. References to and from the moved block are adjusted to retain the original relationship.

The Move command, which you will find at **Edit -> Move** on the main menu, is only enabled directly after a cut operation. Its keyboard short cut is **Ctrl Y**.

Move cannot maintain the logic of the sheet if you move a block to an area which is already occupied.

Delete

Use Delete to clear all the data from a cell or group of cells. There is no "undo" option once done, so **use this with caution**. To use the Delete function select the cell(s) to be emptied, click **MENU** and choose **Edit -> Delete**.

The keyboard short cut for this command is **Ctrl-K**.

Replicate

Use Replicate to make multiple copies of a cell or block of cells. The steps are:

1. Cut or copy the source cells to the clipboard.
2. Select the top left cell of the area where you want the first copy to be placed.
3. Choose **Edit -> Replicate** and the dialogue box will open (Figure 11.2).
4. Fill in the number of replicated blocks you want in each direction.
5. Click on OK.

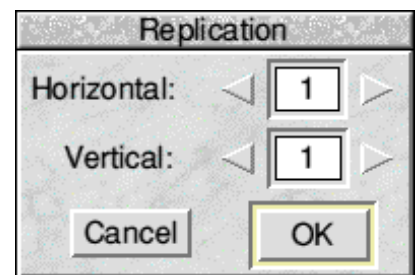


Figure 11.2

The contents of the clipboard are pasted into the spreadsheet the requested number of times.

Alternatively, if you want to fill an area with the contents of the clipboard, select all of that area at stage 2 above. The dialogue box will then appear with the correct number of horizontal and vertical replications already filled in

Replication is useful if you want to copy a block with several different entries. Making copies of a single value or formula is easier using block entry.

Moving references - an exact description

Paste, Move and Paste values commands all handle references in slightly different ways. Also the rules used to move absolute references are different.

The rules are devised to give sensible results in most practical situations, and you won't often need to worry about the details. Nevertheless, you may at some time need an exact description of the rules. Rather than supplying extensive tables of rules, we give a set of examples which take all possible cases into account.

When a block is moved, three types of reference are involved:

- A cell in the block refers to another cell also in the block; this is an internal reference.
- A cell in the block refers to a cell outside the block; this is an outward reference.
- A cell outside the block refers to a cell inside; this is an inward reference.

Each type of reference can be absolute or relative.

Examine Figure 11.3, in which each kind of reference is shown:

	A	B	C
1	37	=A1+1	=@A@1
2	=B2	46	=B2+1
3	=@C@3	81	=@B@3+1
4			
5	Original sheet		

Figure 11.3

Consider shifting block B1...C3. Then the reference types are

- relative internal: C2
- absolute internal: C3
- relative outward: B1
- absolute outward: C1
- relative inward: A2
- absolute inward: A3

Figure 11.4 shows the effect of **cutting** B1...C3 and **pasting** it to C2.

	A	B	C	D
1	37			
2	=B2		=B2+1	=@A@1
3	=@C@3		46	=C3+1
4			81	=@B@3+1
5	Cut B1..C3 and paste to C2			

Figure 11.4

Figure 11.5 shows the effect of **copying** B1..C3 and **pasting** to D2.

	A	B	C	D	E
1	37	=A1+1	=@A@1		
2	=B2	46	=B2+1	=C2+1	=@A@1
3	=@C@3	81	=@B@3+1	46	=D3+1
4				81	=@B@3+1
5	Copy B1..C3 and paste to D2				

Figure 11.5

Figure 11.6 shows the result of a **cut** of B1..C3 followed by a **move** to C2.

	A	B	C	D
1	37			
2	=C3		=A1+1	=@A@1
3	=@D@4		46	=C3+1
4			81	=@C@4+1
5	Copy B1..C3 and move to C2			

Figure 11.6

Finally, Figure 11.7 shows the effect of the **copying** B1..C3 and doing **paste values** to D2.

	A	B	C	D	E
1	37	=A1+1	=@ A@ 1		
2	=B2	46	=B2+1	38	37
3	=@ C@ 3	81	=@ B@ 3+1	46	47
4				81	82
5	Copy B1..C3 and paste values to D2				

Figure 11.7

Techniques 12 - Use of Multiple spreadsheets

Schema2 lets you use two or more sheets in a single application. For example, you might use one sheet to hold a table of relatively fixed data, and another sheet to get data from that table and use it in a calculation. Alternatively, you can arrange your application as a separate sheet for each department in your company, together with a single summary sheet to display the results of all the departments together.

Sheet naming convention

When you use two or more sheets in the same application, the sheets need to refer to one another so as to exchange vital items of data. You need to understand how sheets are named.

Every sheet in a Schema2 application has a name which appears at the top of the sheet. This name is usually but not necessarily the same as the file name (within a directory) in which the sheet is saved.

This is the way that sheet names are created and manipulated:

1. When you make a new sheet by clicking on the Schema2 icon, the name is set to **Untitled**, possibly followed by a number.
2. When you make a new sheet using the **Create New Spreadsheet** dialogue, the name is the one you enter.
3. When you save a sheet for the first time, the suggested name in the save dialogue box is the first 10 characters of the sheet name. You can accept the suggestion, or change the file name if you wish. The name of the sheet is usually reset to be the final section of the file name - that is, the name within a directory.

In this dialogue the field for the sheet name can be up to 40 characters. The dialogue also includes a button labelled **Change name on save**. If you unset this button the name of the sheet will not be changed when you save it so the file name and the sheet name can be entirely different.

Using multiple sheets

The rules for using multiple sheets are straightforward:

1. An application can use any number of sheets up to 50. Each sheet has its own set of views.
2. The sheets in an application must all have distinct names.
3. When you write a formula which refers to a cell in a different sheet, you must precede the cell name by the sheet name, written between a ! sign and a dot. Examples of such references are:
!Heathrow.A1
!Gatwick.B7
4. When a formula refers to a block in a foreign sheet, only the first cell name needs to have the sheet name in front. The second reference may have the same sheet name but this is not essential. The use of a different name is illegal. For example:
Allowed: !Heathrow.B1..C100
!Heathrow.B1...!Heathrow.C100
Forbidden: !Heathrow.B1...!Gatwick.C100

The various spreadsheets in an application work together as a single entity. Normal rules of recalculation apply across all spreadsheets, so that a change in one spreadsheet can trigger an avalanche of changes in many others.

An example of multiple spreadsheet working

With many spreadsheet applications you will find it natural to use several spreadsheets. For example, consider a department store which keeps statistical records on a monthly basis, the spreadsheet for January might look like this:

	A	B	C	D	E	F	G
1	Macchiavelli's Emporium - Records for January 1992						
2							
3	Department	Millinery	Lingerie	Shoes	Hardware	Toys	Totals
4							
5	Sales	£34512	£97003	£17007	£5667	£23760	£177949
6	Cost of stock	£19435	£69446	£12569	£3902	£8566	£113918
7	Wages	£3413	£7834	£6407	£970	£4788	£23412
8	Notional rent	£2100	£4500	£4000	£800	£2500	£13900
9	Profit	£9564	£15223	£(5969)	£(5)	£7906	£26719

Figure 12.2

At the end of the month, this sheet is filed under the name **Jan**, and another spreadsheet with exactly the same format is started for February.

At the end of the year, the managers ask for a summary, and a further spreadsheet is set up. Twelve rows of this sheet contain copies of the Profit rows from each of the monthly sheets. Part of the summary spreadsheet, displaying formulae rather than numbers, is shown in Figure 12.3:

	A	B	C	D	E	F	G
1	Macchiavelli's Emporium Summary for 1992						
2							
3	Department	Millinery	Lingerie	Shoes	Hardware	Toys	Totals
4	January	=!jan.B9	=!jan.C9	=!jan.D9	=!jan.E9	=!jan.F9	=!jan.G9
5	February	=!feb.B9	=!feb.C9	=!feb.D9	=!feb.E9	=!feb.F9	=!feb.G9
6	March	=!mar.B9	=!mar.C9	=!mar.D9	=!mar.E9	=!mar.F9	=!mar.G9

Figure 12.3

This arrangement has the advantage that sheets are kept small and yet any change or correction is reflected in the summary sheet as soon as it is made.

You can make entries in the summary sheet using the block entry method. For example, you can set up row 4 by selecting cells B4 to G4, and entering the formula **=!jan.B9**. The other cells in the row are automatically filled in with properly adjusted versions of the formula.

Here is a second example of multiple sheet working: using a spreadsheet as a data base. This involves only two sheets, but uses them in quite different ways.

The application is designed to keep track of a diet, so that at the end of a day you can easily work out how many calories you have consumed, and also the quantities (in grams) of carbohydrate, protein, fat and fibre.

The first spreadsheet is a data base. It is called **Table** and can be found in the **Examples** directory. It contains a list of all the foods you are likely to eat, together with the calories, carbohydrate, protein and fat content per hundred grams. Such tables can be found in various books on diets. The start of this sheet is shown in Figure 12.4:

	A	B	C	D	E	F
1	Item	Calories	Protein	C'hydrate	Fat	Fibre
2	All Bran	270	14.0%	46.0%	3.0%	24.0%
3	Apple	47	0.4%	11.8%	0.1%	2.0%
4	Baked beans	74	5.0%	12.7%	0.5%	6.0%
5	Baked potato	136	3.9%	31.7%	0.2%	2.7%
6	Bread	215	9.2%	41.6%	2.5%	7.4%
7	Canned tomato	12	1.1%	20.0%	0.0%	0.0%
8	Carrot	30	0.7%	6.0%	0.5%	2.4%

Figure 12.4

A useful function for extracting data from a sheet is MATCH. For example, you can use MATCH to discover how much fibre there is in 100 grams of bread.

MATCH takes three arguments:

- A 'key' value which you want to look up.
- A range, which must be a single row or column, which is supposed to contain the key value.
- Another range, of the same size as the first range.

The function looks up the key value in the first range. If it finds the key, it returns the corresponding value from the second range. If not, it returns the string **Value not found**.

The **Table** sheet holds 41 entries, from row 2 to row 42. To give an example,

=MATCH("Bread",!Table.A2..A42, !Table.F2..F42)

returns the number **7.4**, because **Bread** is the *fifth* entry in range **A2..A42**, and the *fifth* entry in **F2..F42** is 7.4.

Similarly, you would expect

=MATCH("Cement",!Table.A2..A42, !Table.F2..F42)

to give the value **Value Not Found**, since a list of foods won't include cement.

In general, the first parameter to MATCH won't be a constant but the value of a cell entered by the user.

The second spreadsheet of the application, called **Records**, is also found in the **Examples** directory. It is used to record the actual quantities eaten. Each portion you eat occupies a row of the spreadsheet. You enter the name of the food in column A, and the quantity (in grams) in column B. You need to have a pair of scales or be an honest guesser of weights.

The rest of the sheet is pre-filled with formulae which calculate the amount of nutrients you have consumed. For example, every cell in column C has a formula to calculate the number of calories for that item. The formula has to look up the calories in 100gm of the food and multiply by the amount you ate.

The formula uses the MATCH function to look up the name of the food in the table. For example, the entry in cell C2 is

=MATCH(@A2, !Table.@A@2..@A@42, !Table.B@2..B@42) * @B2/100

This formula calculates the calories you have ingested by eating B2 grams of A2 (for example, 35 grams of bread). The factor of 100 allows for the fact that the entries in table are in units of 100 grams.

The rest of columns C, D, E, F, and G are filled with adjusted copies of this formula. The making of part of the references to the tables absolute ensures that the correct cells are referenced.

A segment from this sheet (Figure 12.5) is:

	A	B	C	D	E	F	G
1	Item	Quantity(gm)	Calories	Protein	C'hydrate	Fat	Fibre
2	Muesli	30	111.90	2.55	16.89	3.69	2.70
3	All Bran	40	108.00	5.60	18.40	1.20	9.60
4	Milk	170	102.00	32.81	47.26	16.49	0.00
5	Milk	40	24.00	7.72	11.12	3.88	0.00
6	Orange	185	46.25	0.37	2.22	0.00	0.74

Figure 12.5

The need for Recalculation

A formula which refers to a missing sheet generates the error result **Reference to non-existent spreadsheet**.

Suppose you forget to load some of the sheets in a linked group. When you evaluate the sheets the formulae in them are likely to give a **Reference to non-existent spreadsheet** error. The cure is to load the missing spreadsheet and choose **Utilities -> Evaluation -> Recalculate** all from the main menu.

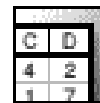
If you have multiple spreadsheets then it is often useful to save a **Workspace** for them (see Section 13). The workspace contains a list of all your sheets, and when you load the workspace all the spreadsheets are brought back for you. This avoids problems with missing sheets.

Techniques 13 - Saving, Loading and Workspaces

This section of the manual is about the various methods you can use to preserve spreadsheets in your filing system and bring them back later.

Saving

When you **save** a spreadsheet, you make a copy of it in the filing system. The copy records everything about the sheet, including cell size information, effects, and any graphics you may have inserted. Later, you can **load** the sheet back into Schema2 and continue where you left off.



C	D
4	2
1	7

A sheet is saved as a Schema2 file, with file type **ASheet**. Every saved sheet has a unique file identifier, which consists of a series of directory names and finally a leaf name. For example, one file may have the file identifier:

ADFS::HardDisc4.\$.techpix.animals

where the leaf name is **animals**.

A loaded Schema2 application knows sheets by their sheet names only. The name of each sheet is displayed in its title bar, and sheets refer to each other using their names. As we explained in section 12, the sheet name is usually but not always the same as the leaf name of the file identifier.

To save a sheet for the first time:

1. Make sure that the directory in which you want to save the sheet is visible on the screen.
2. Click **MENU** on the sheet and open the dialogue from **File -> Save**.
3. If necessary, change the leaf name of the sheet to the one you need. The sheet name is altered accordingly, unless the Change name on save button in the Spreadsheet name dialogue has been cleared.
4. Pick up the file icon with the mouse pointer and drag it to the directory.

The save process may not succeed, there are some things that might go wrong:

- It is easy to drop the icon in the wrong place. If you put it into the wrong directory you can copy the file across to correct matters; but if you drop it somewhere invalid you may have to restart the save process.
- The file system can run out of room. If this occurs, make some more disc space available and try again.

Saving a sheet previously loaded from a file and changed.

If you want to save the new version under a *different* name, follow the same sequence as you would if you were saving a new sheet.

To save the new version under the *same* name, you have a choice of methods:

- Click on **File -> Save** to bring up the dialogue and click on **OK**.
- Click on **File -> Save** menu item, there's no need to open the dialogue.
- Make sure that the sheet to be saved has the input focus (the border will be yellow) and press **F3** followed by **Return**.

File Information

When you are handling large files, you may find it useful to know in advance how much disc space a given sheet will need. To get information about your file, choose File -> Info from the main menu. Schema2 displays a panel like the one shown in Figure 13.1:

The panel shows several items of information, including:

- whether it has been modified since it was created or loaded
- the spreadsheet name
- the full path name (the end may be truncated)
- the size in bytes
- the date the file was written

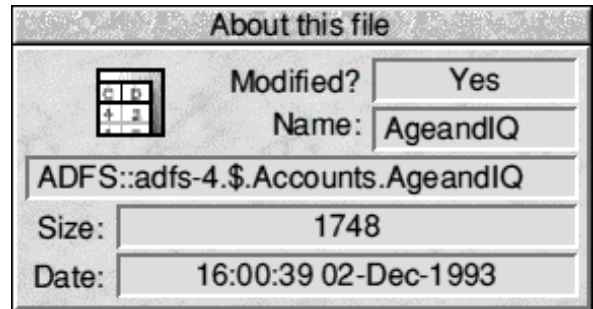


Figure 13.1

The Modification Marker

Another useful item of information can appear in the title bar of every sheet. If you have made any change to a sheet since it was loaded, the title bar shows the **Q** symbol. This implies that you must save the sheet before you finish the Schema2 session - unless, that is, you deliberately decide to discard the new version.

Hints on saving

Schema2 does not save sheets automatically, except when you save a workspace (see the last part of this section). It does warn you when you propose to take some action which destroys an unsaved sheet, so that it is difficult to lose a sheet by mistake.

During a long Schema2 session you should save your sheets at regular intervals to protect yourself against power cuts and other misfortunes. Remember to keep backups of important sheets. One day your disc will crash - it is not a question of **if**, but **when**.

Saving Selection

You can save the selected block of a spreadsheet as a smaller spreadsheet in its own right. To do this:

1. Select the area you wish to save.
2. Bring up the Save dialogue by choosing **File -> Save** in the main menu.
3. Click the Selection button in the dialogue.
4. Change the name of the file.
5. Drag the file icon to the directory.

When you have saved part of a sheet it is handled as if its top left corner were in cell A1.

Loading

Schema2 sheets can be loaded in several ways:

To start a Schema2 session, double-click on the file icon of the spreadsheet. This will start up the Schema2 application and load the sheet.

or

If Schema2 is already present on the icon bar, drag the spreadsheet file icon to the icon bar icon.

or

If a spreadsheet has already been opened, you can drag a file icon to it. The filed spreadsheet (which may be a saved selection of a spreadsheet) is copied into the area marked by the top left corner of the selected block. All cell references in the newly loaded spreadsheet are adjusted to take the new position into account.

To give an example, suppose that you have saved a spreadsheet with 4 rows and 6 columns, called Small.

1. Open a new sheet of size 28 rows by 10 columns.
2. Select cell C15.

3. Drag the file icon for Small to the new sheet.

The contents of Small are copied into cells **C15..G18** of the new sheet. This process will fail if the sheet you load does not fit into its new position.

Loading can fail for several reasons:

- The sheet you try to load (or another one with the same leaf name) may already be loaded.
- Random Access Memory can run out. To make more space delete another application or some unwanted sheets in Schema2.
- The disc may fail with read errors. This is a serious emergency, but panic action may make things worse. **Don't** load your backup disc into the drive which caused the problems, because the malfunction could be in the **drive**, and you could easily end up with a damaged backup disc as well. If you can, make a copy of your backups on another machine before you do anything else; above all, take time to think and get advice.

Workspaces

A workspace gives you a convenient way of storing and retrieving a set of related spreadsheets, macros and associated resources. In particular:

- You can save the current state of the system in a workspace and shut down Schema2. The next day you can restart and return immediately to the original state. This is especially useful if you are using a large number of linked spreadsheets and macros.
- When writing a Schema2 application you can set up a workspace so that a user can just click on the workspace icon to start the application.

Saving a workspace

To save the current workspace, choose the Save workspace entry in the icon bar menu. If any sheet has the input focus, the **Ctrl-F3** key will also bring up the Workspace Save dialogue.



If you are saving a workspace for the first time, enter a name for the workspace and drag the file icon to a directory. The workspace file has a distinctive icon pattern and the file type **SWorkSp**.

If you are saving a workspace which you loaded earlier, it is enough to click on the OK icon in the dialogue, or even the menu entry itself.

If there are any new sheets which have not been saved, Schema2 gives you a warning; it will not save the workspace unless all the new component sheets have been saved.

If any spreadsheets have been altered since they were last saved, Schema2 saves the new versions automatically when it saves the workspace.

It is best to keep your workspace file in the same directory as the sheet files.

Loading a Workspace

To load a workspace, double-click the workspace icon in the file directory. All the spreadsheets in the workspace are reloaded automatically. The views are reopened in the places, and in the same order, as when you saved the workspace.

Example - Load the workspace file SimpleWS in the Examples directory. It loads the two sheets used in the Calorie counting example in the Multiple sheets section.

More details about workspaces

Workspaces are designed so that if you save a workspace in one place and then copy its directory to another place (or another machine) it will still usually work. The way this happens is described below.

A workspace consists of a list of instructions to load spreadsheet (and other) files in the application, together with a description of the views currently on display.

Each file identifier in the workspace can be expressed in two ways:

- Absolute, as a full path name.
- Relative to the location of the workspace file itself.

If a file name is relative, Schema2 substitutes the name of the directory in which the workspace resides.

When you save the workspace, each file is given a relative name if, and only if, it is in the same directory as the workspace or if it is a member, directly or indirectly, of any sub directory in this workspace. To give an example, consider the following file structure (Fig 13.2):

In this structure **WSDir** is a directory which is a child of **ADFS::Disc\$.User**. It contains a spreadsheet called **Sheet2** and a directory call **Subdir**.

Suppose the workspace is recorded in directory **WsDir**. Then **Sheet1** is given an absolute name:

ADFS::Disc\$.User.Sheet1

The other sheets are given relative names, relative to **ADFS::Disc\$.User.WSDir**.

If the directory **WSDir** is moved elsewhere (say to **ADFS::Disc\$.Sheets.WSDir**) then it will still be loaded correctly as the new position will be used for the relative file names. However if the directory is moved to a different machine (or disc) it may be unable to find any spreadsheet files.

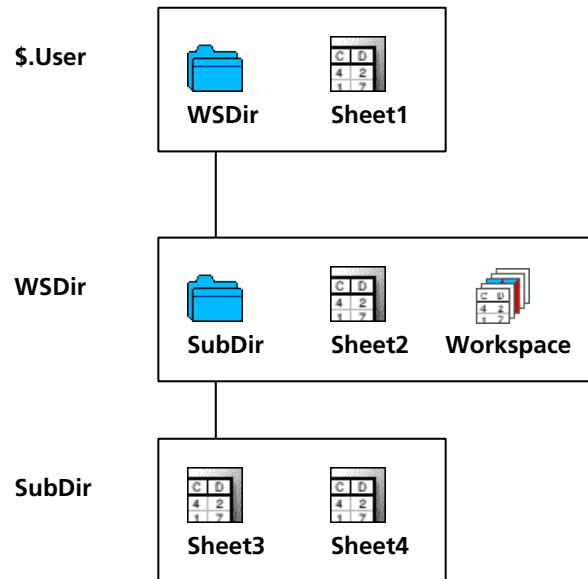


Figure 13.2

When Schema2 locates a workspace it sets the **RISC OS** variable **Workspace\$Dir** to point to the directory.

Resource files

When Schema2 is started, it needs to read some resource files. For example, the arrangement of the menus and of the icon ribbon is stored in various files in a Menus directory, and details of drawings you can put into pictograms are kept in the Pictures directory.

If the directory which contains the workspace file also includes any resource files, these files are used to start Schema2. If a resource file is not found in the workspace, Schema2 uses the corresponding default resource file in the !Schema2 directory.

Resource files let you customise Schema2 by changing the menu structure and icon bar. Further details are given in Section 27.

Note: You should note that a workspace holds the names of files, not the files themselves. This makes it possible for a sheet to belong to two or more workspaces, but it also creates the risk that a sheet could be deleted, leaving a hanging reference in a workspace file. If a sheet listed in a workspace file cannot be found when the workspace is loaded, a warning message is displayed.

Techniques 14 - Importing spreadsheet data

This section explains how to take data produced by other applications and enter it into a spreadsheet.

Computers must deal with many different kinds of data: images, programs, text, lists of numbers, and so on. **RISC OS** recognises this fact by giving each sort of data its own file type - **Text**, **Drawfile**, and so on.

File types for import

Schema2 can import data of six different types, which are:

Drawfiles - A **Drawfile** is a record of a drawing, with elements such as lines, polygons, curves and sprites. It is a standard file type for RISC OS machines.



Drawfiles are produced by the **!Draw** application, and by other applications which specialise in image handling, such as scanner support software, ray tracers, map drawing programs and graphics packages.

The file type of a Drawfile is **DrawFile**.

TSV - A **TSV** file (**Tab-Separated Values**) consists of a sequence of data items, expressed in characters, separated by **Tab** or newline characters. The structure of TSV files is detailed in Appendix C.



TSV files do not have their own file type but are files of type **Text**.

TSV files are often used by word processors to export and import tables of data. They may be referred to as text files by the wordprocessor.

Note that a TSV file does not carry any information about effects such as font, character size or colour.

The file type of a TSV file is **Text**.

CSV - A **CSV** file (**Comma Separated Values**) consists of a sequence of data items separated by commas or newline characters. The items are mainly numbers. Items can also be labels enclosed in double quotes. The structure of CSV files is detailed in Appendix C.



The **CSV** format is used mainly to store and exchange **numerical** data, such as the points needed to plot a graph. CSV files are widely imported and exported by spreadsheet, database and graphing applications amongst others.

The type of a CSV file is **CSV**.

SID - The **SID** (**Software Independent Data**) format is used to record information read from data loggers. A **SID** file can be imported into a spreadsheet in a form suitable for data analysis and graph plotting.



The type of a **SID** file is **SID**.

WK1 - **WK1** files are used for transferring complete spreadsheets between Schema2 and other spreadsheet programs.



WK1 is a widely used standard for spreadsheets. It was originated by Lotus 1-2-3, but is used (or at least understood) by most other commercial spreadsheets.

The file type of a **WK1** file is **WK1**.

Schema - **Schema** files are used to record spreadsheets by the **Schema** spreadsheet application, which is now obsolete. Note that **Schema** and **Schema2** files do not share the same format.



The filetype of a Schema file is **Sheet**.

Importing Sprites and Drawfiles

You can load a Drawfile into an individual cell of a spreadsheet. You may find it useful to put the Drawfile into a **big cell** of suitable size.

The way a Drawfile is displayed in a cell depends on the settings of the **Effect -> Graphics** dialogue. If you set Fit to cell, the image is scaled to fit the cell exactly. Otherwise only part of the image may be shown.

This feature of Schema2 can be used to decorate spreadsheets with company logos or other symbols which make the sheets more readable.

To import a Drawfile and scale it to fit a big cell:

1. Select a block of cells in the sheet and press **Ctrl F** to make a big cell.
2. Drag the file icon of the image to the spreadsheet window.

You cannot load a sprite file directly into Schema2, but you can drag a sprite to the **IDraw** application, which makes it into a Drawfile which contains nothing but the sprite. You may then load this Drawfile into Schema2.

Importing TSV files

When a TSV file is imported each value is stored as a string constant or a number if it conforms to the accepted number layout. The Tab separated values are arranged in rows, and each newline character marks the start of a new row.

There are two ways to import a TSV file:

Drag the Text file icon to the Schema2 icon. This will open a new sheet of the right size and fill it with data from the TSV file.

or

Select a cell in an existing sheet and drag the Text file icon to the sheet. The data from the file is copied into the sheet, using the marked cell as the top left corner of the block. The previous contents of the cells into which the data is loaded are discarded.

If the sheet is not big enough to contain the TSV file, an error is reported.

A common use for TSV transfers is to add marginal totals or averages to tables of numbers embedded in text documents. To do this:

1. Use a word processor to export the table in TSV format, using a command such as **Export as Text**.
2. Import the table into Schema2.
3. Calculate the totals and averages you need.
4. Re-export the sheet to the word processor by choosing **File -> Export-> TSV**

Importing CSV files

There are two ways to import a CSV file:

Drag the CSV file icon to the Schema2 icon. This opens a new sheet of the right size and fills it with data from the file.

or

Select a cell in an existing sheet and drag the CSV file icon to the sheet. The data from the file is copied into the sheet, using the marked cell as the top left corner of the block. Data already in the block before the transfer is destroyed.

If the sheet is not big enough to contain the CSV file, an error is reported.

You can use a CSV file to import information from data bases, so that you can calculate with the data.

Importing SID files

Schema2 uses a supporting application **!ConvSID** to help it read SID files.

To import a SID spreadsheet, take the following steps:

1. Ensure that **!ConvSID** has been seen by the filer.
2. If there is some spare memory (about 100K) you can load Schema2 and drag the SID file directly to the iconbar icon. This opens a new sheet and loads the file. You can also drag the SID file icon directly to an open sheet in which a cell has been selected. The imported sheet is loaded using that cell as origin.

If you are short of space, you will need to translate the file from SID to Schema2 format in a separate operation. Don't load **!Schema2** yet; if you have already loaded it, delete it from the icon bar. Load application **!ConvSID** and drag the SID file icon to the **!ConvSID** icon. When the Save box appears, drag the icon to a suitable directory. This new file is a translation of the SID file into Schema2 format. Quit from **!ConvSID** and double-click on the new file icon; this loads **!Schema2** and sets up the data as a spreadsheet.

Importing WK1 files

You can import a complete spreadsheet from Lotus 1-2-3 or a compatible program provided that you observe the minor restrictions given in Appendix D

If the WK1 file is in **MS-DOS** format, read the last part of this Section (page 183 Importing MS-DOS files) before you start the import process.

Schema2 uses a supporting application **!ConvWK1** to help it read WK1 files.

To import a WK1 spreadsheet, take the following steps:

1. Using the other spreadsheet program, record your sheet in WK1 format.
2. Ensure that **!ConvWK1** has been seen by the filer.
3. If there is plenty of spare memory (about 200K) you can load Schema2 and drag the WK1 file directly to the Schema2 bar icon. This opens a new sheet and loads the file. You can also drag the WK1 file icon directly to an open sheet in which a cell has been selected. The imported sheet is loaded using that cell as origin, and all the cell references are adjusted accordingly.

If you are short of space, you will need to translate the file from WK1 to Schema2 format in a separate operation. Don't load **!Schema2** yet; if you have already loaded it, delete it from the icon bar. Load application **!ConvWK1** and drag the WK1 file icon to the **!ConvWK1** icon. When the Save box appears, drag the icon to a suitable directory. This new file is a translation of the WK1 spreadsheet into Schema2 format. Quit from **!ConvWK1** and double-click on the new file icon; this loads **!Schema2** and sets up the translated spreadsheet.

Sometimes the translation process may find a feature which it cannot translate correctly. If this occurs you see a warning.

Importing Schema Files

Schema2 uses a supporting application **!ConvSch1** to help it read Schema files.

To import a Schema spreadsheet, take the following steps:

1. Ensure that **!ConvSch1** has been seen by the filer.
2. If there is plenty of spare memory (about 200K) you can load Schema2 and drag the Schema file directly to the Schema2 bar icon. This opens a new sheet and loads the file. You can also drag the Schema file directly to an open spreadsheet in which a cell has been selected. The imported spreadsheet is loaded using that cell as origin, and all the cell references are adjusted accordingly.

If you are short of space, you will need to translate the file from Schema to Schema2 format in a separate operation. Don't load **!Schema2** yet; if you have already loaded it, delete it from the icon bar. Load application **!ConvSch1** and drag the Schema file icon to the **!ConvSch1** icon. When the Save box appears, drag the icon to a suitable directory. This new file is a translation of the Schema spreadsheet into Schema2 format. Quit from **!ConvSch1** and double-click on the new file icon; this loads **!Schema2** and sets up the translated spreadsheet.

Certain styles used in Schema may not be rendered in exactly the same way in Schema2. This is most likely to happen to a style which represents positive numbers, negative numbers, zeros and strings in widely different ways.

When you have converted a Schema application into Schema2, you may also notice some minor differences in cell layout, such as the exact spacing between the edge of a cell and its contents. This is due to essential differences between Schema and Schema2, and is entirely normal.

Importing Schema Workspaces

To transfer a Schema workspace to Schema2, take the following steps:

1. Set up a new directory for the application.
2. Load Schema2.
3. If there is a macro file in the Schema workspace,
 - a) Copy this file into the new directory .
 - b) Drag the copied file to the Schema2 icon.
4. List the Schema workspace file. It contains the names of all the spreadsheets in the application,
5. Load each of these sheets into Schema2 and save it into the new directory. Do not discard it.
6. Save the current workspace into the new directory.
7. Run the application to ensure that no problems arise.

Using the clipboard to import data from other applications

Schema2 shares a clipboard with other applications. The data in the clipboard is likely to be one of the following:

- a TSV file from a word processor
- a CSV file from a data base
- a Drawfile

To import clipboard data:

1. Enter the other application and cut or copy the data you wish to transfer.
2. Select a cell in an open Schema2 sheet and choose Edit ç Paste. The data in the clipboard data is imported, using the selected cell as the top-left origin.

Importing data in MS-DOS format

Sometimes you may need to import data which has been recorded in **MS-DOS** format. The data types are most likely to be **TSV**, **CSV** or **WK1**.

Make sure that the file is showing the right filetype. To change the type in **RISC OS 3** you must select the file and the press **MENU** over the directory viewer and choose **File -> Set type** in the filer menu. Type in the filetype required and press **Return**.

Techniques 15 - Exporting spreadsheet data

This section follows on from Section 14, and explains how to export spreadsheet data to other applications.

You can export the following types of data:

- **Drawfiles**
- **CSV files**
- **TSV files**
- **WK1 files**

These types are described in Section 13.

You can also export **Formulae** and **Print text**.

Before you try to export a file of any sort, you should be sure that you have a destination for it. This could be another application window, an application icon or a directory window.

If you are exporting to the **MS-DOS** environment, the destination window should be the directory of a disc formatted for **MS-DOS**.

Exporting Drawfiles

To export a sheet or part of a sheet as a Drawfile



1. Open the main menu and choose **File -> Export -> Draw**.
2. When the Export dialogue appears, mark the Selection box if you only want to save the selected part of the sheet.
3. Change the name of the file if you want to file it.
4. Drag the icon to its destination, which could be a directory window, an application icon such as !Draw, or the window of another application such as a word processor.

This process transfers the whole sheet, or the selected part of it, as an image.

Many of the illustrations in this manual were made in the way we have just described.

Exporting CSV files

The steps to export a file in CSV form are the same as those for a Drawfile. You must choose the main menu item **File -> Export -> CSV**.



CSV files are used to send data to graph-drawing applications and data bases.

Exporting TSV files

The steps to export a file in TSV form are the same as those for a Drawfile. You must choose the main menu item **File -> Export -> TSV**.



TSV files are used mainly to send data to word processors and DTP applications.

Exporting WK1 files

A spreadsheet can be exported in WK1 format and transferred to another spreadsheet program or computer, such as the ACORN Pocketbook machine. This lets you continue work on the sheet even when a **RISC OS** machine is not available.



If you plan to export a sheet in WK1 format, observe the restrictions listed in Appendix C.

Schema2 uses a supporting application **!ConvS2WK1** to export WK1 files.

To export a spreadsheet:

1. Ensure that **!ConvS2WK1** has been seen by the filer.
2. Make sure there is a window to which you can drag the translated file.

If you have plenty of memory -

3. Choose the main menu item **File -> Export -> WK1**. When the Export box appears, do the same as for a Drawfile.

If you are short of memory -

3. Save the sheet in normal Schema2 format.
4. Remove Schema2 from the icon bar.
5. Load application !ConvS2WK1.
6. Drag the Schema2 sheet to the !ConvS2WK1 icon.
7. When the Save box appears, drag the icon to its destination window.

The WK1 file can now be loaded into its target program.

If the translator finds any items it cannot handle, it displays a warning message.

We strongly recommend that before transferring spreadsheets between Schema2 and other spreadsheet programs, you read Appendix D carefully.

Exporting Formulae

To export as **formulae** choose **File -> Export -> Formulae** in the menu and use the **export** box.



Schema2 produces a **text** file with a list of the non-empty cells and their formulae for the whole spreadsheet or the selection. This file can be dragged to the printer application icon to print it.

Exporting Print text

To export as **Print text** choose **File -> Export -> Print text** in the menu and use the Export box.



Schema2 produces a **text** file according to the current set-up for **draft printing** for the spreadsheet. The file can be printed as 'fancy text'. For more information see Section 16 on printing.

Transferring files to the MS-DOS environment.

To transfer any kind of data to **MS-DOS**, you must start with a disk which has been formatted to **MS-DOS** standards. **RISC OS 3** will format such discs directly, but with **RISC OS 2** you will need to use a utility program. The alternative is to format the disc on your nearest **IBM PC**.

Once the disc has been formatted, then under **RISC OS 3** you can record files on it without any formality. These files will be readable by **MS-DOS**. With **RISC OS 2** you will need a utility like **!PCDIR** to look after the disc for you. Start up the utility and drag the files to be saved into its window.

With CSV and TSV files, observe the notes on new lines in Appendix C.

Techniques 16 - Printing

This section deals with spreadsheet printing.

Printing only works properly if you have installed your printer correctly. This is not an easy task, and many owners choose to have their printers installed by dealers or other people with suitable experience. If you do decide to do the job yourself, have the printer manual handy, and read the chapter on printers in the Welcome Guide before you start.

Printing often gives rise to problems which seem to be caused by Schema2, but which are actually due to bad installation of the printer.

Schema2 lets you print in two different modes: **draft** and **image (normal)** print.

Draft printing

Draft printing uses the printer's own character set to get a fast, simple record of a spreadsheet. When you print a sheet in draft, there is no attempt to preserve the appearance of the sheet on the screen. There are no colours, no special fonts, and no graphical information. Cell boundaries are drawn with minuses and vertical bars, and each column is made exactly as wide as it needs to be. An example of draft printing is shown below (Figure 16.1):

```

-----
| |      A      | B | C |      D |      E |
-----
| 1 |Month      |March|June |September|December|
| 2 |Income:     |    |    |    |    |
| 3 |Ice Cream Sales |34000|68000|    95500|    23000|
| 4 |Expenses     |    |    |    |    |
| 5 |Raw materials  |12580|25160|    35335|    8510|
-----
| 6 |Wages        |12500|22000|    25000|    12500|

```

Figure 16.1

Image (normal) printing

Image printing gives an exact copy of the screen version of a sheet, like the one shown in Figure 16.2:

	A	B	C	D	E
1	Month	March	June	Septembe	December
2	Income:				
3	Ice Cream Sales	34000	68000	95500	23000
4	Expenses:				
5	Raw materials	15300	30600	42975	10350
6	Wages	10500	24500	24500	10500
7	Rent	5000	5000	5000	5000
8	Vechicles	2000	4000	4000	2000
9	Profit	1200	3900	19025	-4850

Figure 16.2

The only differences between a screen picture and a printed image are:

- the input area is not shown.
- the screen has a resolution of 45 dots per inch, but the resolution of most printers is somewhat higher - at least 300 dots per inch for a laser or inkjet printer. The quality of a printed sheet is usually much better than that of a display on the screen.
- Many printers are limited to black and white and will display other colours as various shades of grey.

The choice of printing format depends on your type of printer.

Although dot matrix printers are now regarded as obsolete for most purposes they are still sometimes used for printing large spreadsheets, mainly because they can print on continuous paper and their running costs are extremely low. Most matrix printers run fast on draft printing, very slowly on image printing, so they would normally only be used in draft mode.

In practice it may make sense to use draft or low-resolution image printing when you are developing a spreadsheet, and keep the best resolution for the final copy.

Laser printers are just as fast on image printing as on draft. There is therefore no point in using draft printing.

Before you print a sheet (either in draft or in image mode) you must make sure that the print modules are loaded into the machine. To do this run the printer manager program **!Printers**.

If you are short of memory you can save about 180K bytes if you quit the manager after starting it. This will leave the print modules installed so that image printing still works. For direct draft printing, you need the printer manager to be present.

If you want to use draft printing but don't have enough space for the manager, take the following steps:

1. Run Schema2 without the printer manager, and save the sheet as a printing text file (choose **File -> Export -> Print text** from the main menu).
2. Quit Schema2, load the printer manager, and drag the saved file to the printer icon.

The Print dialogues

To call the **Print dialogue**, shown in (Figure 16.3) on the right, choose the **File-> Print...** entry in the main menu, or click on the Print icon in the icon ribbon.

To display the Print Options dialogue, click on the Options button in the Print dialogue. This dialogue is logically part of the Print dialogue, but handles items which are not so often needed. It is shown on the next page (Figure 16.4).

Use the print dialogues to describe the way you want a spreadsheet printed.

Saving options without printing

To save a set of options without printing, select the options you require and click on **Save**. The options will be stored but the spreadsheet will not be printed. Page boundaries are recalculated and displayed.

Printing a spreadsheet

To print a spreadsheet directly, select a set of options and click on the **Print** button. The options you have chosen are remembered so that next time the spreadsheet is printed they will be used by default.

Draft

Turn this button on to print in **draft mode**; otherwise printing is in **image mode**.

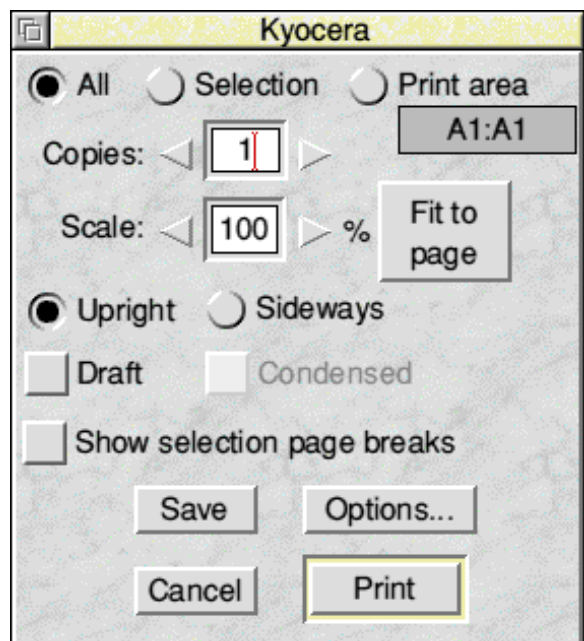


Figure 16.3

Common Print options

The options in this group apply to both styles of printing.

All/Selection - A pair of radio buttons controls this option. Choose All to print the entire spreadsheet, or Selection to limit the display to the cells in the selected block.

Copies - This group of entries controls the number of copies printed. You can edit the number directly, or adjust it up or down by clicking the two arrow boxes at the sides of the main icon.

Show Selection page breaks - This button controls whether page breaks in printing are shown on the screen.

Options for draft mode

The options in this group apply only to draft mode and have no effect on image printing.

Condensed - Some dot matrix printers offer a 'condensed' mode, in which characters are squeezed up to much less than their normal size. If you select this option, draft printing uses the **Condensed** mode (if any) and can fit a wider spreadsheet on to a single page.

Characters per line - This entry lets you determine the maximum number of characters per line. Adjust the figure, depending on whether you are using condensed printing.

There is an upper limit of 255 characters per line.

If you specify a number which your printer cannot support, the end of each line may be missing, or the line may be 'folded' so that it runs on to the next line of paper. In either case the printed output will be a mess.

Options for the image mode

The options in this group only work for the image mode.

Collated - This option is useful when you print more than one copy of a spreadsheet of several pages. If collation is **On** each copy is printed in order, ready for binding. If it is **Off**, all the copies of each page are printed together.

Consider a four-page document being printed in six copies. With collation, the order of output would be

1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4

Without collation it is

1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4

On most laser printers, each page image is only set out and transmitted once, no matter how many copies are made at the time. Uncollated printing is usually faster than collated.

Scale - You can print the spreadsheet on any scale from 25% to 200% of its nominal size. This scale is independent of the current magnification of the spreadsheet when on display.

Upright/Sideways - You can print the spreadsheet either **Upright** on the page ((known as *Portrait* orientation) or **Sideways** on the page (known as *Landscape* orientation). The difference between the two is shown by Figure 16.3. A spreadsheet which takes two pages in portrait mode may fit onto a single landscape page.

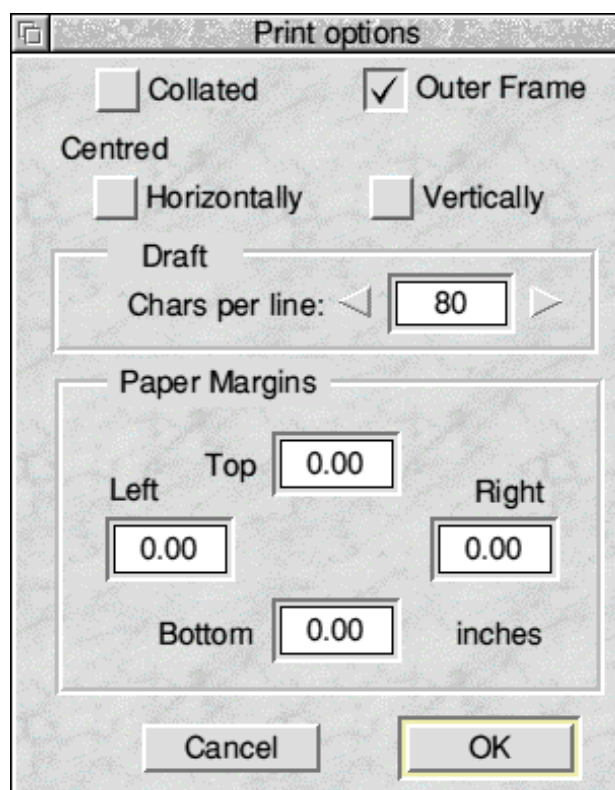


Figure 16.4

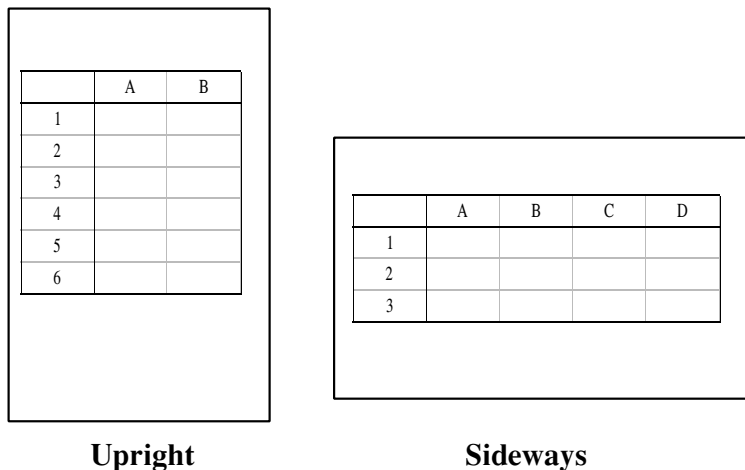


Figure 16.3

Fit to page - This button attempts to fit a spreadsheet, or the selected part of the spreadsheet, on to a single page. The button selects the orientation (upright or sideways) which makes best use of the page and sets the scale so that the page is filled, margin to margin, in at least one direction.

The Fit to page button fails, and reports an error, if:

- The scale needed to fit the spreadsheet on a single page falls below 25% or above 200%.
- Any row or column in the display has a forced page break.

Margins - The Print Options dialogue lets you set up the margin for each of the edges of the page. The units (cm, mm, inches or points) are those you chose in the Choices dialogue in the icon bar menu.

If all the margins are set to zero, the system uses the default margins set when the printer is installed.

If any of the margins is non-zero, the set of margin figures in the dialogue over-rides the default setting.

A zero figure corresponds to the physical edge of the page. Bear in mind that most printers can only get within a few millimetres of the edge.

It is useful to set your own margins if you expect to bind the spreadsheets into book form, and need a specially wide margin on one side of the page.

Centred - There are two options to determine where on the page the spreadsheet is drawn.

If **Centred Horizontally** is on the sheet is printed midway between the two vertical margins, otherwise it is aligned to the left margin.

If **Centred Vertically** is on the sheet is printed midway between the upper and lower margins, otherwise it is aligned to the top margin.

Selecting areas to print

If you ask for a spreadsheet to be printed, the default action is to print the whole spreadsheet except for the input area. However, there are various ways to limit the amount which is displayed:

Selecting a block - Select a block and choose the Selection option in the Print dialogue. Only the selected block is printed.

Omitting the standard Column and Row labels - Hide column and row labels by choosing the **View -> Labels** entry in the main menu and labels are not printed. Any rows or columns you have locked still appear.

Omitting the grid - Hide the grid by clicking on the **View -> Grid** entry in the main menu, and the grid of lines separating cells is omitted.

Black grid - To print the grid in black instead of grey select **Black grid** in the Choices dialogue, which is described in Section 26. You can display this dialogue by selecting Choices on the icon bar menu,

The black grid option is provided mainly for use on matrix printers, which often fail to render grey lines effectively. It does not affect the colour of the lines on screen.

Hiding rows and columns - You can hide groups of rows or columns from view, so that they don't appear either on the displayed version of the spreadsheet or in the printed version.

To hide a group of one or more columns:

1. Select the columns.
2. Choose **Edit -> Columns -> Hide columns**.

The columns disappear, but are still present in the spreadsheet, so that formulae which involve their contents remain valid.

You can hide any number of groups of columns, at any time.

To bring all columns you have previously hidden back into view choose **Edit -> Columns -> Reveal columns**.

Rows can be hidden and brought back in exactly the same way, except that the menu entries are **Edit -> Rows -> Hide rows** and **Edit -> Rows -> Reveal rows**.

Hiding rows and columns is useful when you don't want to reveal all the information in a spreadsheet. If you hide the system labels and use locked rows and columns instead, it is impossible to tell that anything has been hidden.

Controlling spreadsheet layout

When you prepare a large spreadsheet for printing, it is useful to know about and to control the page divisions in advance.

When a spreadsheet is displayed on the screen, you will normally see that some rows and columns are bordered by solid black lines instead of the usual grey ones. These lines represent **page breaks**, and are calculated from the print options saved with the spreadsheet. If no printer driver has been loaded, a dummy set of values is used.

Note that when a spreadsheet is small enough to go into a single page, no page breaks will be shown.

Page breaks are recalculated automatically every time you do something to change the layout of the spreadsheet - hiding a column, switching off labels, or changing the height of a row.

Normally, the system arranges the page breaks so that as many rows and columns as possible are placed on one page. Normal sized cells are never split across pages, but big cells may be. You can tell in advance because the page break line cuts across the big cell.

In ordinary conditions, the page breaks are worked out on the assumption that the whole spreadsheet is to be printed. If you set the Show Selection page breaks option, the page break lines are set assuming that you only want to print the selected area, and change every time you alter the current selection. This is useful in planning, but does make the system a little slower.

You can control the position of page breaks by attaching a 'page break flag' to any row or column. If present, this flag forces a page break immediately to the right of the column or below the row.

To set or unset a page break flag, select a row or column, and choose one of the entries:

Edit -> Row -> Page break
or
Edit -> Column -> Page break

Date and Time (Temporal) values

Dates and times are normally handled internally as numbers, known as **temporal values**. Two methods of measuring time (**time modes**) are used. They can be calculated by adding or subtracting the number of elapsed seconds or days (as appropriate) to a known value:

- The **Schema** mode measures the number of seconds. The time 00:00:00 on 1st January 1990 has the value: **62793187200**
- The **Lotus** mode measures the number of days. The time 00:00:00 on 1st January 1990 has the value: **32874**

Example:

The time value for 16:00:00 on 30 December 1989 is in Schema mode:

$$62793187200 - ((24+8) * 60 * 60) = 62793072000$$

in Lotus mode:

$$32874 - (1 + (8/24)) = 32872.66666666$$

In practice such calculations rarely have to be made.

Entering Dates

Working out and typing huge number is an inconvenient way to enter a date. Schema2 has a feature to simplify the entry of dates and times. If you attach a date format to a cell and enter a value to that cell Schema2 expects you to type a date. To get the right value you must follow some simple rules:

1. Enter the 'components' of the date in the same order as the corresponding components in the format string. The default order is determined by the **Schema2\$DateFormat** environment variable but is usually day-month-year.
2. Type the day as a number in the range 1-31.
3. Enter the month in any of the following ways:
 - as a number in the range 1-12
 - as a roman number in the range I to XII
 - as a full name in the range January-December
 - as a three-letter abbreviation in the range Jan. to Dec.
4. You may use upper or lower case letters anywhere.
5. Enter the year either as a 4-figure number or as a 2-figure number, when the century is assumed to be the current one. To enter a date in the first century use a four-figure number such as 0065.
6. If necessary, the components of the date should be separated by spaces, dots, slashes or any other symbol to avoid ambiguity.

Examples:

If the order is day-month-year, the system will accept any of the forms

210893, 21.8.1993, 21 viii 93, 21AUGUST1993, 21 Aug 93

Schema2 will reject a 'date' which is ambiguous or non-existent, including

11 11 1, 29 Feb 1993

If your date format has a format string with the components in any other order, then that is the order which applies.

Schema2 does not accept dates in terms of day-numbers or week-numbers.

Entering Times

If you give a cell a time format, then Schema2 can accept a time for that cell as raw data. The default order of components (determined by the variable **Schema2\$TimeFormat**) is normally:

hour - minute - second - am/pm

When you type a time you must state the hour, but the other components are optional. The numbers must either be two figures long, or be separated by other characters; you can write AM or p.m. - in capitals or lower case; separating the letters with dots is optional.

Example

Acceptable times are

9.00, 145422 , 2359 , 1.30am, 4:15PM

When you enter a time into a cell, it is converted into a temporal value on the **first** day of the calendar. This makes it possible to add date and time temporals and obtain sensible results.

Strings

A string is a segment of text entered into a cell. Strings are used to identify rows, columns and other entries in the sheet.

Examples: George Snooks Cumulative Profit

If a string is too long for a cell, it can spill over into any adjacent cell which happens to be empty. See Section 7 for a discussion on this topic.

Block designations

A block designation is the name of a block, either in the current sheet or in another sheet. Block designations can be entered thus:

=A3..C9

or

=CD4><FE56

Cells which contain block designations can be used instead of explicit blocks in other formulae. Block designations are included under the general heading of constants because they are unaffected by the contents of any other

Techniques 18 - Operators

Formulae in Schema2 consist of **operands**, **operators**, **brackets**, **macro calls** and **calls to standard functions**. In this section we discuss the action of operators.

The operators in a formula indicate how the values of the operands should be combined.

Example In the formula: **=A1+1**

A1 and **1** are operands and **+** is an operator, which shows that the value in cell A1 and the number 1 must be **added** to work out the formula.

Arithmetic operators

The arithmetic operators are:

Operator	Alternative form	Action
+		Addition
-		Subtraction
*		Multiplication
/		Division
%	MOD	Modulus
^		Power

With certain exceptions, all these operators take arguments which may have any signed value between 10^{300} and 10^{-300} . The results can have any value in the same range.

Special rules are:

- Division by zero is forbidden.
- The remainder operator % expects integer operands. If either of the operands is a mixed number it is truncated to the nearest integral value towards zero. Remaindering by zero is forbidden.
- The power operator cannot raise a negative value to a fractional power.

Examples

=11 MOD 5	gives 1
=45 % 16	gives 13
=3 * 3	gives 27 (3*3*3)
=2 ^ 0.5	gives 1.4142... (SQRT(2))
=67.54 % 9.754	gives 4 (67 remainder 9)
=23/0	gives Division by zero
=-3.456 ^ 0.39	gives Bad operand for power operator
=10e200 * 10e200	gives Number too big

String operator

There is one string operator, CAT, the concatenation operator. It takes two strings and joins them together.

Example

= "Hello" CAT " " CAT "World" gives "Hello World"

Relational operators

The relational operators compare two values to see if they are the same or if one is greater than the other. If the relationship is found to be **true** they return **1**, otherwise they return **0**.

The relational operators are:

Operator	Action
=	Equals
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal

If the comparison is between two strings the comparison is based on a case sensitive alphabetic ordering (ASCII ordering).

Examples

	=11 > 5	<i>gives 1</i>
	=12 <> 12	<i>gives 0</i>
	="AGE" <> "age"	<i>gives 1</i>
	="Age" > "Beauty"	<i>gives 0</i>
	= 101 < 37	<i>gives 0</i>
but	= "101" < "37"	<i>gives 1 (why?)</i>

The NOT operator

The **NOT** operator takes a single operand. If its value is zero it returns 1; otherwise it returns 0.

NOT is used to turn round relations.

= NOT (A1 > B1)

means the same as

= (A1 <= B1)

Bitwise logical operators

The bitwise logical operators operate on 32 bit signed two's complement binary integers.

The bitwise logical operators are:

Operator	Alternative form	Action
	OR	Or
&		And
**	NEQ	Exclusive Or
	INVERT	Invert bits

All these operators take two operands except for INVERT, which takes 1.

The operators act on each bit of the number independently. Their actions are determined by these truth tables:

A	INVERT A
0	1
1	0

A	B	A B	A&B	A**B
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Examples

```

is      =5 AND 12
        0101 AND
        1100
        0100    (4 in decimal notation)
        = INVERT 0    gives -1

```

Bitwise shift operators

The bitwise shift operators operate on 32 bit signed binary integers, like the bitwise logical operators. The bitwise shift operators are:

Operator	Action
>>	Logical shift right
<<	Logical shift left

Each shift operator shifts its first operand right or left by the number of places given by the second.

Examples

```

is      = 11 >> 1
        01011 >> 1
        00101    (5 in decimal notation)
        = 12 << 2    gives 48

```

IF ... THEN ... ELSE ... ENDIF

The conditional operator, IF ... THEN ... ELSE ... ENDIF, returns one of two values depending on another value. If the operand between the IF and the THEN is not zero, then the value between the THEN and the ELSE is returned. Otherwise the value between the ELSE and the ENDIF is returned.

Examples

```

= IF 0 THEN "Hello" ELSE "Goodbye" ENDIF    gives "Goodbye"
= IF 10>5 THEN 10 ELSE 5 ENDIF              gives 10.

```

Conditional operators can be nested, as in:

```

= IF A1 < 100 THEN IF A1 < 50 THEN "AWFUL" ELSE "MEDIUM" ENDIF ELSE "GOOD" ENDIF

```

Avoid using too many levels of nesting as this can make expressions and difficult to understand and obscure their meaning.

Using multiple operators and precedence

A formula can use more than one operator.

Examples

```

=5 + 12 - 20    gives -3
=10 * IF "Life">"Death" THEN 70 + 5 ELSE 0
ENDIF    gives 750

```

The order in which the operators are used is defined by the rules of precedence. In general the higher the precedence, the sooner an operator is used. The table on the right gives the precedence of the operators (higher precedence first). All the operators on the same line have the same precedence.

Examples

```

=5 * 10 - 12 / 2    gives 44 ( * and / have higher precedences than - )

```

Precedence	Operator
10	IF ... THEN ... ELSE ... ENDIF
9	NOT INVERT -(initial minus)
8	^ (power)
7	* / % (MOD) CAT
6	+ -
5	>> <<
4	= <>
3	< > <= >=
2	& (AND)
1	(OR) ** (NEQ)

`=10>5 AND 5 = 6` gives 0 (> and = have higher precedences than AND)

Some expressions contain chains of operators of the same precedence. These chains are evaluated from left to right if the operators take two operands, but from right to left if they take one operand. For example,

`a*b/c` associates as `(a*b)/c`
but

- NOT INVERT A6 associates as `-(NOT(INVERT(A6)))`

Brackets

In order to change the order of calculation from the one specified by the **precedence** table you must use the brackets (and). Anything within the brackets is calculated first. Brackets must always come in pairs, otherwise you will get the error **Bracket mismatch**.

Examples

`=(5 * (10 - 12)) / 2` gives -5, not 44
`= A1 > 18 AND A1 < 40` gives 1 if A1 is between 18 and 40, else 0

Techniques 19 - Sheet variables

A sheet variable is an arbitrary name for a constant, an expression or a block. Sheet variables are useful in building large sheets, since they help to make expressions more readable. Here are some examples:

- You can give names to numerical and string constants, such as

VAT	for 0.175
PI	for 3.14159265
DNA	for "Data not available"

Once you have defined these names you can use them in formulae instead of typing out the equivalents in full.

- Many of the expressions in a sheet may use common sub-expressions, like `(100*A2/(A2+B2))` and `(100*B2/(A2+B2))`. You can assign these expressions to sheet variables (called perhaps PERCENT_GIRLS and PERCENT_BOYS) and then use these variables instead of the expressions. When either of the cells A2 or B2 are changed the variables are updated with the rest of the sheet.
- You can attach a variable name to any block and use that variable as a block parameter to any standard function. For example, you can take the area G1..G90 and call it RESULTS. You could then write

`=AVERAGE(RESULTS)`

There are four main operations you can carry out with a sheet variable:

Creating Sheet Variables

To create a sheet variable you choose a name and a value or expression. The name must start with a letter, and continue with letters, digits or the underscore sign. However:

- the name may not exceed 15 characters in length.
- the name should not be the same as any existing name in the same spreadsheet,
- the name should not clash with the name of a cell; for example **HG567** would not be a valid name,
- the name should not clash with the name of an operator, a standard function or a system macro.

For example:

AND, CELL, CRM_WIDTH

would all be invalid names.

In user names, upper and lower case are not distinguished.

The **value** of a sheet variable can be anything which is allowed in a normal cell, such as numeric or string constants, expressions and block names.

To create a new sheet variable:

1. Bring up the main menu and choose **Edit -> Variable -> New**.
2. Enter the name of the new sheet variable and press **Return**.
3. The name of the new variable now appears in the input window, in the icon normally used to display the name of the current cell. Enter any legal value or expression as the value of the sheet variable, and press **Return**. If you click on the Tick icon, you continue to edit the sheet variable.

Editing Sheet Variables

To edit a sheet variable:

1. Open the main menu and choose a variable from **Edit -> Variable -> Edit**.
2. The name of the variable now appears in the input window. Change the expression for the variable.
3. To enter and continue editing, click on the Tick icon, or to enter and return to editing cells press **Return**.

Deleting Sheet Variables

A sheet variable can only be deleted if it is not referred to anywhere in the application.

1. Choose the name of the variable from the **Edit -> Variable -> Delete** submenu on the main menu.

Using Sheet Variables

When you enter an expression, you can use a sheet variable name instead of a constant, a cell name, or a block. You can refer to a sheet variable in a different sheet if you precede its name with a suitable foreign sheet reference, like this:

!Physics.Mean

Sheet variables behave just like invisible cells with names instead of standard identifiers like **A1** or **G7**; this is a useful clue to understanding how they work.

Techniques 20 - Standard Functions

This section gives an overview of Schema2 standard functions and of two standard macros which behave as if they were standard functions. Full details of all these can be found in the appendix on **Schema2 Standard Functions**.

A standard function is a request for a special computation. Standard functions are used in formulae; for example, the formula:

$$= 5 + \text{SQRT}(7)$$

includes a call to the SQRT standard function and calculates 5 plus the square root of 7. It works out to be 7.64575...

Standard functions cover various fields of application, such as statistics, mathematics, and finance. Examples of standard functions you have already used are **SUM**, **COS** and **MID**.

Arguments and results

An **argument** of a standard function is a quantity which it needs to do its job. In our example, 7 is the argument of the **SQRT** function.

Arguments are often called *parameters*. The two words mean the same.

Every standard function has a fixed number of arguments. Many functions, especially those taken from the sphere of mathematics, take only one argument. For example **COS**, **SQRT** and **RAND** take one argument each. Other functions take two or more arguments depending on the type of function.

One function - **NOW** - takes no arguments at all.

An argument can be any of the following:

- a number
- a string - that is, a sequence of characters
- a block - that is, a rectangular area of a sheet

In a function call, a parameter can be either a constant or a formula which yields a value. For example, suppose that Cell **A1** contains the number **4.5**, and cell **B1** holds the string "**CRM Ltd**". Then a number argument could be written as any of

45 **A1** **A1+7**

a string argument could be any of

"Schema2" **B1** **B1 cat " wrote " cat "Schema2"**

a block argument could be

A1..B1

The set of arguments for a function is always enclosed in round brackets. If there is more than one argument, you must write the arguments in the correct order, with commas as separators.

If there are no arguments (as with the standard function **NOW()**) you must still include the brackets.

Some examples of formulae with function calls are :

=TAN(45)	(Tan takes one number argument)
=GAUSS(G6,7)	(GAUSS takes two number arguments)
=LEFT("Object", 3)	(LEFT takes two arguments - a string and a number)

Every standard function produces a result. This can be a number, a string or a fault report such as **Argument must be positive or zero**.

You can enter the names of standard functions using the mouse. This technique is described in Section 2.

The Function Groups

The functions are arranged into various groups, depending on their areas of application. Some of the groups are quite specialised; you won't understand them unless you have the appropriate background knowledge.

The function groups are listed below:

Time functions

Schema2 stores dates and times as **temporal values** (see Section 17). The time functions work with temporal values, and make it easy to convert between temporals and conventional dates and times. There are also some functions to move a temporal forward by a month or a year.

The functions include:

SECONDS	MINUTES	HOURS
DAYS	MONTHS	YEARS

to convert a temporal into the components of a standard time or date

TIME

to turn a temporal into a string like **1993/08/29/14/46/47**

TTIME

to convert a standard date or time into a temporal

MONTHSADD	YEARSADD
------------------	-----------------

to advance a temporal by some months or years

NOW

to find the temporal value of the moment of evaluation

Note: Schema2 has two **time modes**, **Schema** and **Lotus**, which are alternative ways of representing dates and times. You can select the mode by using the Utilities ç Evaluation ç Values dialogue in the main menu. It is *very important* to set the convention **before** you enter any temporal values into the sheet, as changes do not work retrospectively.

If you have been consistent, all the time functions work equally well in either convention.

String functions

This group includes a set of functions to manipulate strings, to convert between characters and their ASCII values, and to search any string for the occurrence of a specific sequence. The functions included are:

LEFT	MID	RIGHT
-------------	------------	--------------

to extract portions of strings

LEN

to find the length of a string

LOWER	UPPER
--------------	--------------

to convert strings to lower and upper case, respectively

FIND

to find a given substring in a string

ASC	CHR
------------	------------

to convert between strings and ASCII codes

Scientific Functions

This group carries a conventional name (we don't really think these functions are specially dear to scientists). The group includes:

SINH COSH TANH

Hyperbolic functions

EXP LN LOG

Antilog, natural logarithm, and log to the base 10

SQRT

Square root

Trigonometric functions

This group comprises the standard ratios of trigonometry and their inverses:

SIN COS TAN COT

Normal ratios

ASIN ACOS ATAN ACOT

Inverse ratios

ATAN2 ACOT2

Safe versions of ATAN and ACOT; they take two arguments and work correctly for angles such as 90 degrees. In general, **ATAN2(x,y)** gives the same result as **ATAN(x/y)** but doesn't fail when y = 0.

DEG RAD

Conversion between degrees and radians

By default, these functions work in degrees but you can switch to radians if you bring up the main menu and open the dialogue box you will find at **Utilities -> Evaluation -> Values**. Then select the **Radians** radio button.

You can also use this dialogue to switch back to degrees.

Array functions

Each array functions uses a block of cells as its parameter, and calculates a number which depends in some way on all the values in the block. For example:

=SUM(A1..J8)

adds together all the 80 values in cell block **A1..J8**, and delivers the result.

The array functions include

SUM PRODUCT

to add or multiply all the cells in a block

MAX MIN

to find the largest or smallest value

COUNT

to count the number of non-zero values in the block

ALL

true if all the values are non-zero

AVERAGE STDEV ESTDEV

measures of the mean and dispersion

Random number generators

Schema2 offers two random number generators: one to produce numbers with a rectangular distribution, and one which offers a gaussian or normal distribution.

The random functions include:

RAND GAUSS

Algebraic functions

This group includes functions often used in simple algebra:

ABS

absolute value

INT FLOOR CEIL

nearest whole number

ROUND

round a value to a given number of decimal places

SIGN

-1, 0 or +1 depending on sign of argument

Financial functions

The functions in this group help with computations connected with compound interest. For example, there is a function which can answer the question:

If I borrow £100000 at an interest rate of 15%, and repay £15500 each year, how many years will I have to pay?
(The answer is a little over 24 years)

The financial functions are of special interest to people who borrow or lend money, or manage other people's wealth. The full list of financial functions in Schema2 includes:

FV	IPMT	IRR	NPER
NPV	PMT	PPMT	PV
RATE	SLN		

Statistical functions

The statistical functions let you fit a straight line or an exponential curve to a set of data points in two dimensions. The functions can deliver the slope and the intercept of such a line, and make predictions on the basis of the fitted curve. The functions are:

GRADIENT INTERCEPT

find parameters of best fit line or exponential curve

PREDICTION

predict a value on the basis of the best fit line or curve

Matrix functions

Schema2 has a full set of matrix functions for addition, subtraction, multiplication, and transpose. There are also functions to initialise matrices and to clear them, and to solve simultaneous linear equations of any order.

In Schema2 a matrix is held as an array or block of cells. The matrix functions take arrays or blocks as their parameters, and uniquely among standard functions, write their results directly into the sheet.

The matrix functions include

	MATCLEAR	MATINIT
initialise a matrix	MATDIAG	MATCOPY
	MATADD	MATSUB
matrix arithmetic	MATMULT	
	MATTRANS	
transpose a matrix		
	MATSOLVE	
solve a system of linear simultaneous equations		

Standard macros

Two standard macros, MATCH and SEARCH, allow values to be extracted from a table. Full details of these macros are given in part 13 of Appendix A.

Other standard macros which may be useful include database operations, more financial functions and some string operations. They are all described in parts 14-16 in Appendix A on Functions and Macros.

Techniques 21 - The Graph dialogues

Introduction

In this section we describe how to use the graph definition dialogues. The actual graph types are discussed in Section 22, decoration of graphs is described in Section 23, and their disposal and storage in Section 24.

We do not explain here how to set up new types of graph; this topic is covered in the volume on Macros.

A graph is a pictorial representation of data already present in a sheet. Schema2 offers a selection of graph types, suitable for data of different sorts. You can plot each graph type in many variants of colour, size, 'depth', and selection of axes.

For simple applications Schema2 has a general graph plotting function with a built-in expert system. The function examines the data and chooses a suitable graph type by following a set of rules.

The data you plot does not need to be in a contiguous area of the sheet; for example, you can draw a scatter diagram based on any two columns.

Specifying a graph

To draw a graph with Schema2, you can call up several dialogues. In broad terms, you would use

- New Graph to select the type of graph and start graphing
- Graph Data to choose the data to be plotted
- Graph Axes to set up the graph axes
- Graph Options to set the graph size

However, you may not have to use all these dialogues for every graph, as Schema2 makes sensible default decisions by itself. In the simplest case you don't have to use *any* of the dialogues.

Plotting a simple graph

This option bypasses all the dialogues and lets Schema2 make its own decision about how to plot the data.

1. Select a block which contains the data to be plotted. If there are strings in the top row or left-hand column of the block, they will be used as labels.

2. Choose **Utilities -> Simple graph** from the main menu. Alternatively, can type **Ctrl T** or click on the Simple Graph icon in the icon ribbon.

Schema2 examines your data and may draw one of -

- a histogram
- a line chart
- a scatter diagram

Schema2 displays a warning message if it can't find a suitable form of chart for the data.



New Graph dialogue

To select a particular type of graph, use the **New Graph** dialogue. To open this Choose **Utilities -> Graphs** on the main menu, or you can press **F7** or click on the Graph icon in the icon ribbon. This is shown on the right in Figure 21.1.

The **Graph Type** field shows the current graph type. To change it, click on the menu button and choose the type from the Graph Type menu.

The buttons in the **Graph Parameters** area bring up the other graph dialogues. Open up the ones you need (if any), set your parameters, and then click on Graph to draw your graph.

Cancel closes the dialogue without any effect.

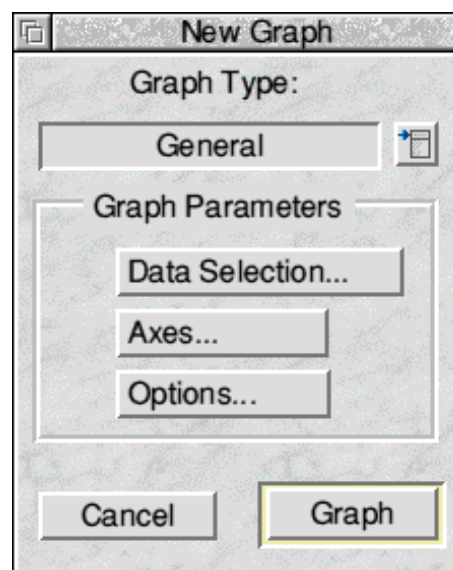


Figure 21.1

Graph Data Dialogue

To display this, shown in Figure 21.2 below, click on the **Data Selection** button in the New Graph dialogue.

Most graphs use one independent data set, usually plotted on the horizontal axis, and one or more dependent sets, usually shown on the vertical axis.

The fields in this dialogue are:

Graph Data block - This is purely informative and gives the position of the block selected for graphing.

Data Direction - This determines how the data sets are read from the selected block and if they are displayed as rows or columns. The default selection assumes that the data sets are stored along the **longer** dimension of the selected block - rows in a short wide block, or columns in a high narrow one. However, you can change the orientation by clicking on one of the radio buttons.

X data set - This field lets you choose the independent data set, which is usually plotted along the X axis. There are two options:

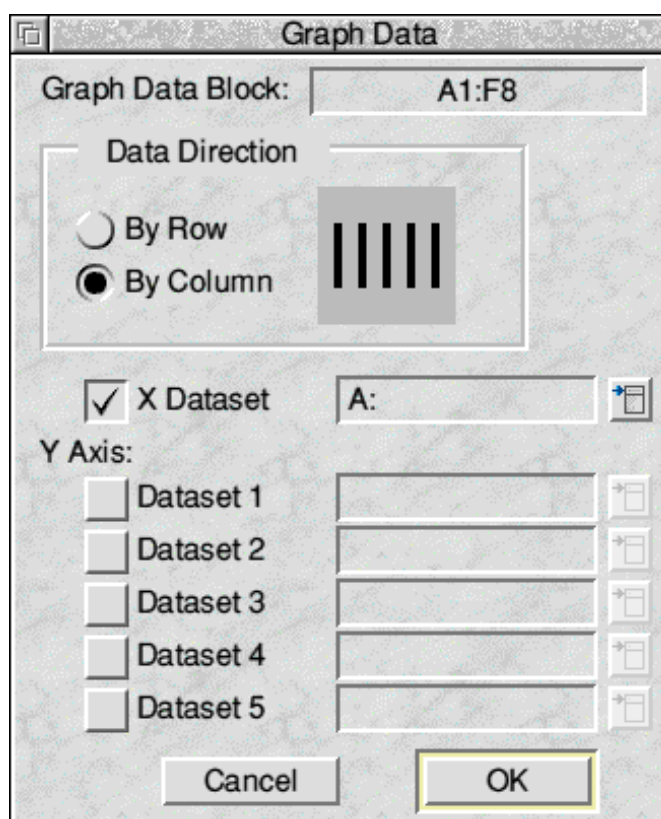


Figure 21.2

- You can leave the X data set out completely by clicking the On/Off button. If the data set is omitted a dummy set with the values 1,2,3... is supplied automatically if needed.
- If you decide to use an X data set, you can specify any row or column in the selected block. Open the pop-up menu from the button on the right and choose a data set.

The menu will show the row or column numbers and the row or column labels if they are present at the edge of the selected block.

Y axis - This lets you choose up to five dependent data sets in the same way as the X data set. These need not occupy adjacent rows or columns in the selected block but can be selected in any order, from any position. For example, if the selected block includes columns A to Z, the data sets can come from columns C, R and M (in that order).

Cancel - This cancels the selection and closes the Graph Data dialogue.

OK - This confirms the selection and closes the Graph Data dialogue.

Default selection: If the Graph data dialogue is not called at all, then:

- If there is only one data set in the selected block, it is taken as the Y data set, and a dummy X data set is supplied automatically.
- If there are at least two data sets, the first one is taken as the X data set, and the others, up to five, are used as the Y data sets.

Graph Axes dialogue

To display this (Figure 21.3) click on the **Axes** button in the New Graph dialogue.

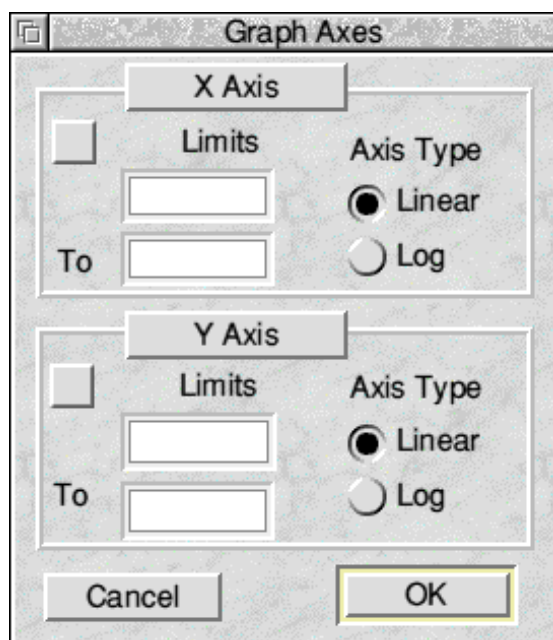


Figure 21.3

You can use the Graph Axes dialogue for any graph which has at least one numerical axis. Each axis can be set with the following attributes:

Limits - For each axis you can switch on limits by clicking the button on the left. You can then enter the range in the two writable icons.
If the limits are active, they set the extent of the axis and data points outside are ignored.
If no number is supplied for any limit the value is taken as zero.

Axis type - Either axis can be plotted on a linear or logarithmic basis.

Cancel - This cancels the selection and closes the Graph Axes dialogue.

OK - This confirms the selection and closes the Graph Axes dialogue.

Default selection: If the Graph Axes dialogue is not used at all, no axis limits are used, and both axes are linear.

Graph Options dialogue

To display this dialogue, click on the **Options** button in the Graph Type dialogue (Figure 21.4).

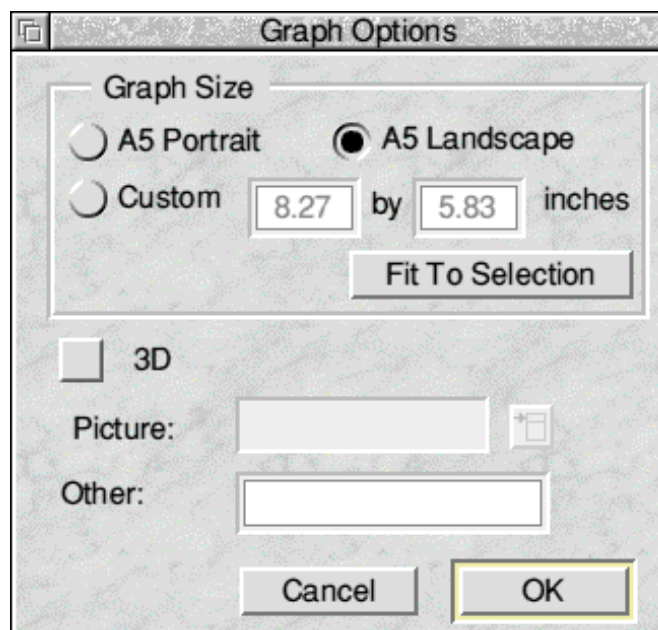


Figure 21.4

The Graph Options dialogue is used for several purposes. You can set the overall graph size and appearance.

A5 Portrait - The graph will fit on to a standard A5 sheet with the long side vertical. Note that A5 is exactly half of A4.

A5 Landscape - The graph will fit on to an A5 sheet with the short side vertical.

Custom - The graph will be exactly the size given in the two numerical fields which specify the horizontal and vertical dimensions, respectively. You can set these to any values you like.

Fit to selection - This option is useful when a graph is to be written back into a spreadsheet (see Section 24). To use the option, alter the current selection on the sheet to mark the place you want the graph to go, and then click the button.

Note that the area you choose must be empty. It will be made into a big cell automatically.

Alternatively you could drop a graph into a Big cell ensuring that the **Effect -> Graphics** dialogue is set to your requirements.

3D - This button ensures that if its type is appropriate, the graph will be drawn with a solid or 'three-dimensional' appearance. This is for appearance only, a 3D graph does not contain any more information than a flat one.

Picture - Pictograms are drawn from a selection of Drawfiles to be found in a dedicated sub directory, which is usually **Pictures** in the **!Schema2** directory. To select a Drawfile for a pictogram, click on the menu button to the right and choose a name from the menu.

Other - This area is not needed for any of the standard graph types but may be used to supply special parameters to other types of graph.

Cancel - This button cancels the current selection and removes the Graph Options dialogue.

OK - This button confirms the selection and removes the Graph Options dialogue.
Default selection: If the Graph options dialogue is not used at all, the default graph size is A5 (landscape), 3D is not enabled, and the picture is the first one in the dedicated sub directory.

Techniques 22 - Basic Graph types

This section describes the various graph types you can use with Schema2. New types can be added to the basic set by writing new graph macros.

Schema2 displays graphs in full colour, and assigns different colours to various sections of data, although this manual is printed in black and white so this cannot be seen.

Line plots (often called X-Y plots)

The line or X-Y plot diagram is used to show numerical data where one or more 'dependent' data sets are plotted against an independent one.

The following sheet (Figure 22.1) contains data about the number of traffic deaths per 100,000 persons in each of four states of the USA.

	A	B	C	D	E
1	Year	New York	Connecticut	Massechusetts	Rhode Island
2	1951	13.9	12.9	10.2	8.0
3	1952	13.7	10.7	10.1	8.2
4	1953	14.5	12.7	10.8	8.2
5	1954	13.2	10.7	10.5	7.6
6	1955	13.7	14.1	11.2	10.1
7	1956	13.7	13.0	10.8	8.2
8	1957	13.6	12.8	10.3	8.3
9	1958	13.4	10.4	11.4	8.4
10	1959	13.3	10.0	10.8	8.6

Figure 22.1

A line graph drawn for the selection A1..E10 produced Figure 22.2:



Figure 22.2

Marked lines

A marked line plot is similar to a line plot except that the data points are identified with markers. This can be useful when a graph is to be printed and the colours of the various lines are not distinct, as in this manual.

Ways of changing size and appearance of markers are discussed in Section 23.

The road death data, plotted as marked lines, produced Figure 22.3:

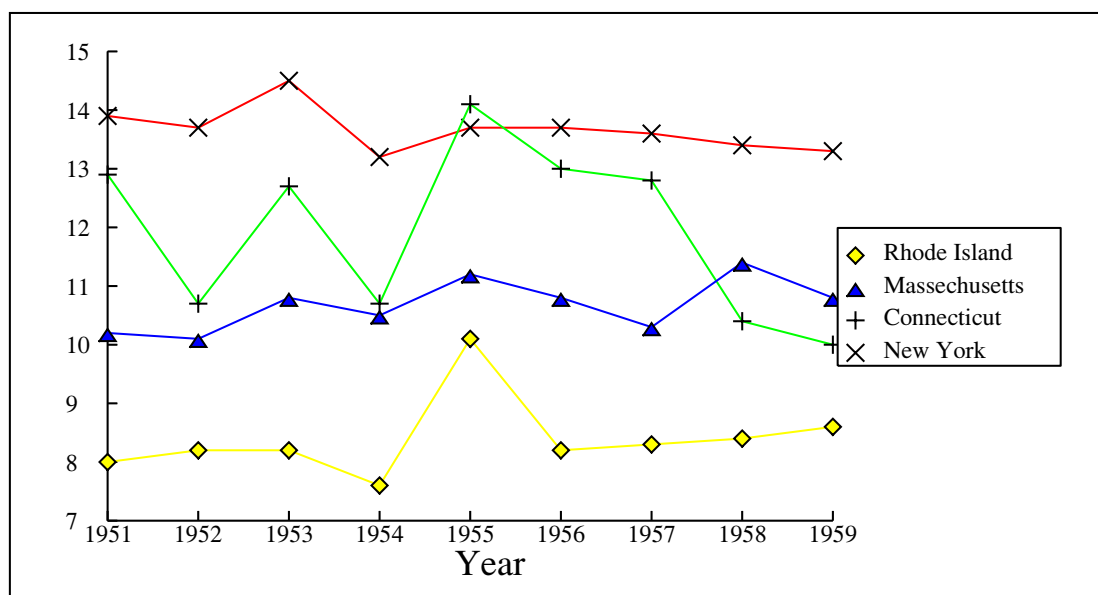


Figure 22.3

Scatter diagrams

Scatter diagrams are useful when neither of the two variables to be plotted is independent, but both are measurements of related functions.

The following graph (Figure 22.4) shows the relationship between the weight of an organism and its age in days at reproduction. Both scales are logarithmic

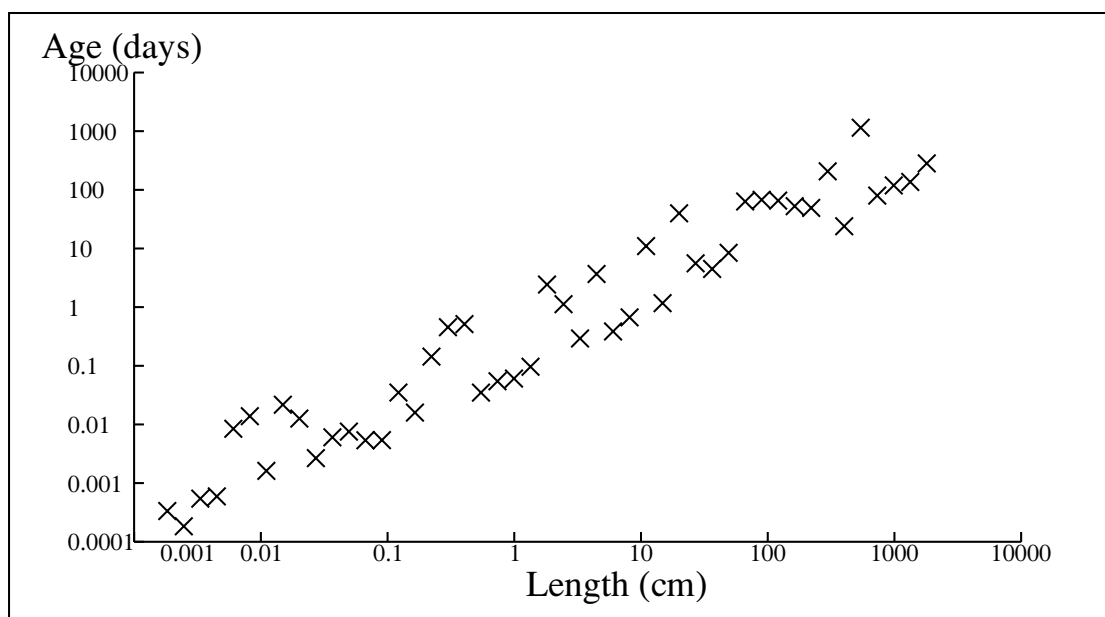


Figure 22.4

Vertical histograms

The following sheet (Figure 22.5) shows the relative GNP per head of the main Chinese provinces in 1991

	A	B
1	Province	GNP per head (China=100)
2	Shanghai	372
3	Beijing	297
4	Tianjin	223
5	Guangdong	262
6	Ningxia	82
7	Sichuan	75
8	Gansu	73
9	Guangsi	70
10	Guizhou	62

Figure 22.5

A vertical histogram plotted on this data produces Figure 22.6:

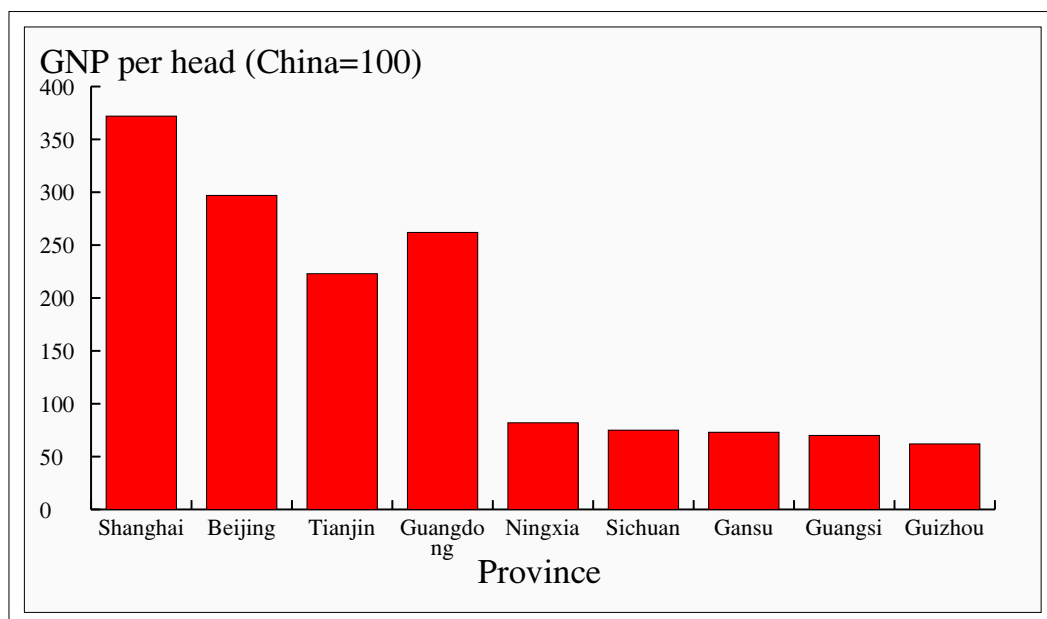


Figure 22.6

This diagram highlights a potential difficulty in placing the labels for vertical bars. If a label is too long to fit into one line, the graph plotter will break it in two. If possible the break is made at a space or punctuation mark, but it could come anywhere. Labels which do not fit into two lines are truncated.

Vertical histograms can also display several variables, which appear as columns side-by-side in different colours. Consider the table in Figure 22.7:

	A	B	C	D	E
1	Food	% Carbohydrate	% Protein	% Fat	% Water, etc
2	Bread	72	12	3	13
3	Butter	12	1	85	2
4	Cheese	20	33	40	7
5	Apples	78	4	0	18
6	Beans	34	44	2	20
7	Bullseyes	100	0	0	0

Figure 22.7

This can produce the graph in Figure 22.8:

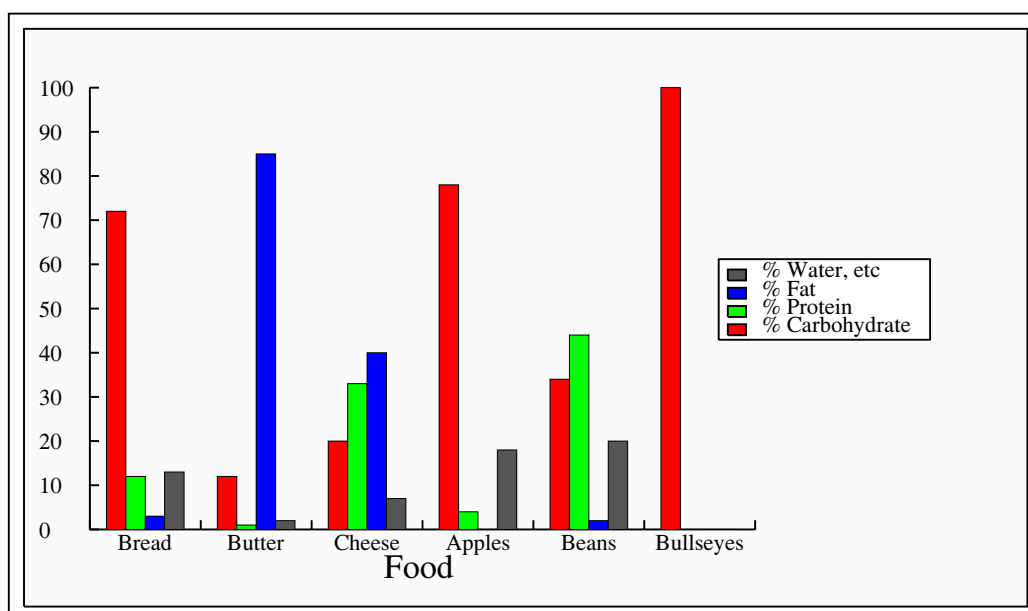


Figure 22.8

Stacked Vertical histograms

Stacked vertical histograms are used when it is sensible to add variable values together and display them in the same column. For example, the following graph, based on the same data as the one just above, shows how the nutrient contents of various food add up to 100% (Figure 22.9):

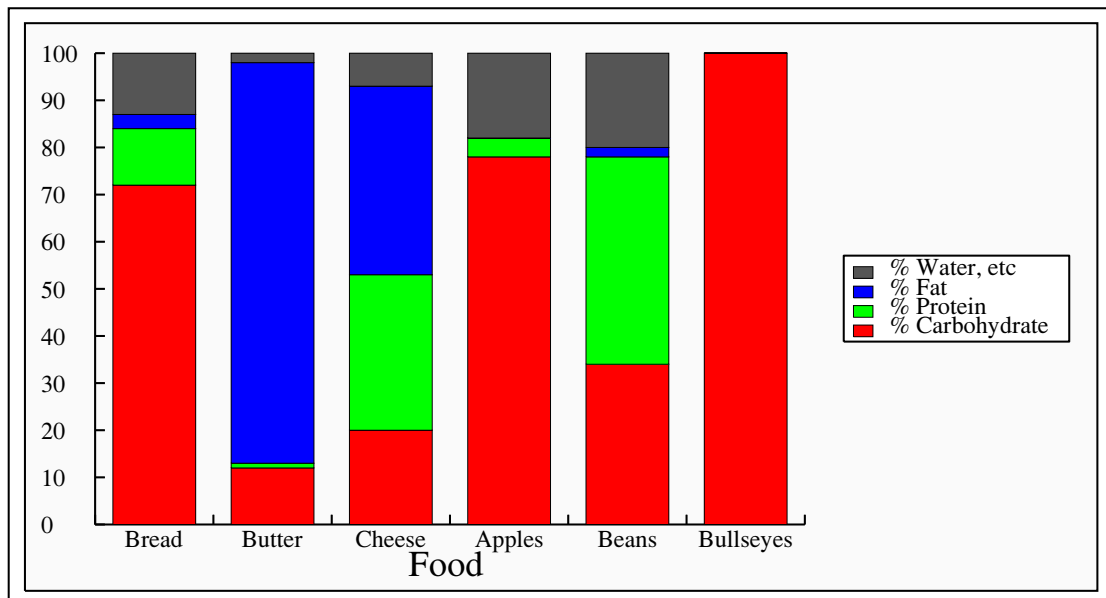


Figure 22.9

Horizontal histograms

A horizontal histogram is basically a vertical histogram turned on its side. The X-data set is plotted vertically, and bar labels are displayed on the left rather than underneath the chart. This allows room for more labelled bars than the vertical histogram.

A horizontal histogram is shown below in Figure 22.10. Compare it with Figure 22.6, which is based on the same data:

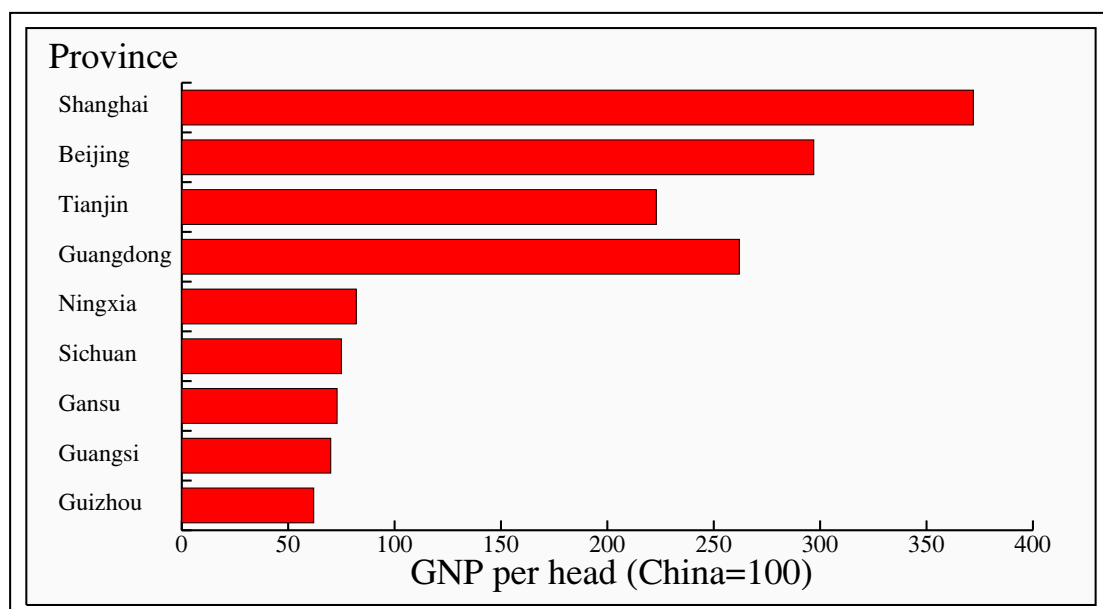


Figure 22.10

Stacked Horizontal histograms

Horizontal histograms can also be stacked. For example, Figure 22.11:

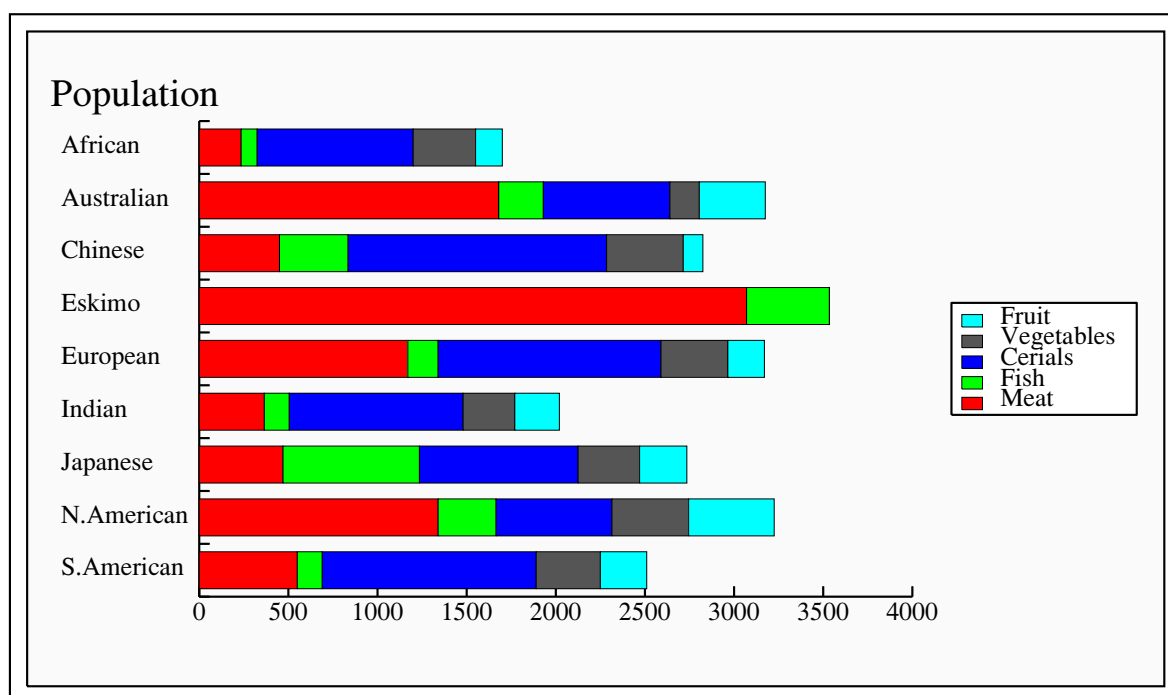


Figure 22.11

Pie Charts

A pie-chart is an alternative way of showing data where the various numbers add up to a fixed total, such as 100%.

A typical example, based on block A1..E2 of the data in Figure 22.7, is shown in Figure 22.12:

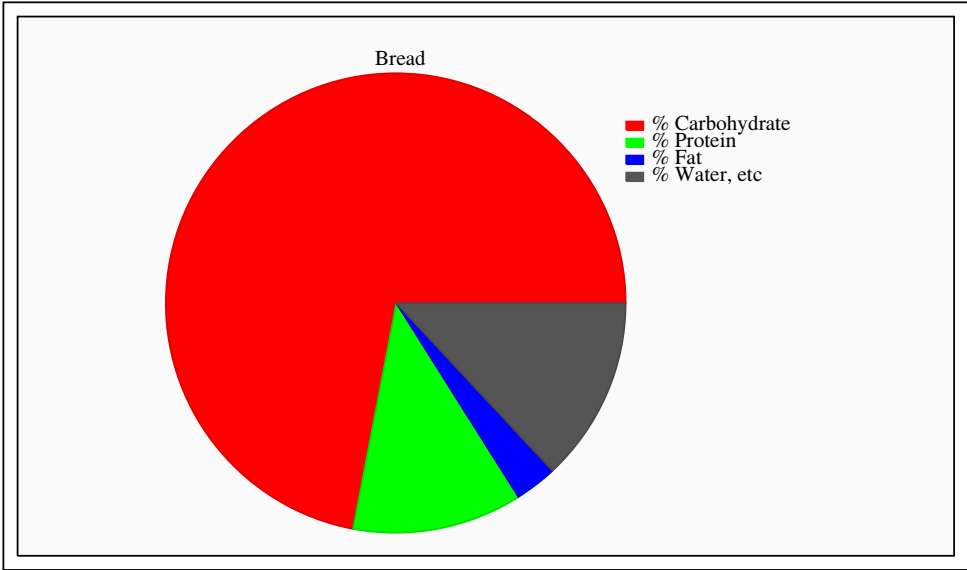


Figure 22.12

Several pie charts can be drawn on the same sheet. Figure 22.13 is based on block A1..E7 of the food data, with the data sets taken by row:

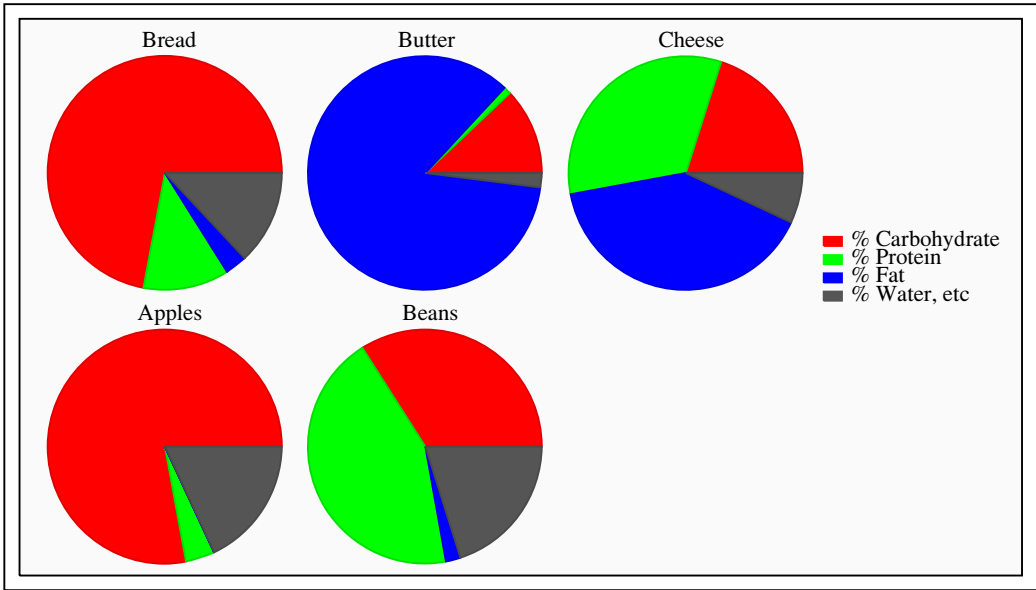


Figure 22.13

Pictograms

This spreadsheet (Figure 22.14):

	A	B
1	Company	Locomotives in service
2	LMS	453
3	LNER	790
4	GWR	212
5	SR	344
6	London Underground	237

Figure 22.14

produces the chart in Figure 22.15:

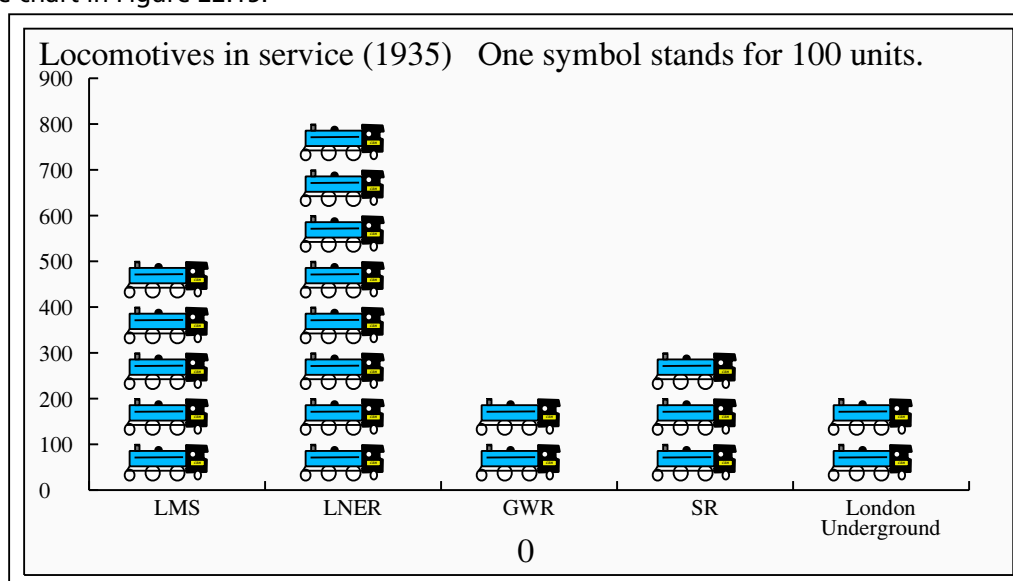


Figure 22.15

Note that the 'train' Drawfile was selected using the picture option in the Graph Options dialogue.

The graph scales the data so that not more than 10 pictures appear in any column. Values are then rounded to the nearest (scaled) unit.

Pictograms can also be drawn using a different image for each data set. Schema2 can select the right image from the picture file provided that the label in the sheet is exactly the same as the name of the file in the **Pictures** directory within **!Schema2**.

The Drawfiles are kept simple so that redrawing is at an acceptable speed. You can use more complex Drawfiles but this will slow the redrawing of the sheet. If you want to use more than one image in a data set then the Drawfiles **must** be in the **!Schema2.Pictures** directory.

The standard set of pictures includes the following:

People: Baby, Boy, Couple, Gentleman, Girl, Man, Woman,

Animals: Bee, Bird, Cat, Cow, Dinosaur, Dog, Eagle, Elephant, Fish, Frog, Horse, Lion, Penguin, Sheep, Snake

Plants: Flower, Tree

Transport: Bike, Boat, Bus, Car, Caravan, Engine, Taxi, Train, Plane

Instruments: Drum, Keyboard, Trumpet, Violin

Miscellaneous: Book, Church, Cross, Football, House, Radio, School, Skull, TV

If you wish to extend your pictogram library, hundreds of images are available, either as public domain software or as commercial collections. You can design your own pictograms with the **!Draw** application.

The sheet in Figure 22.16 is used to produce a pictogram made of several images:

	A	B
1	ANIMAL	Number seen
2	dog	4
3	sheep	3
4	cow	4
5	eagle	6
6	penguin	7
7	dino	7
8	frog	6
9	fish	4

Figure 22.16

This produces the pictogram in Figure 22.17.

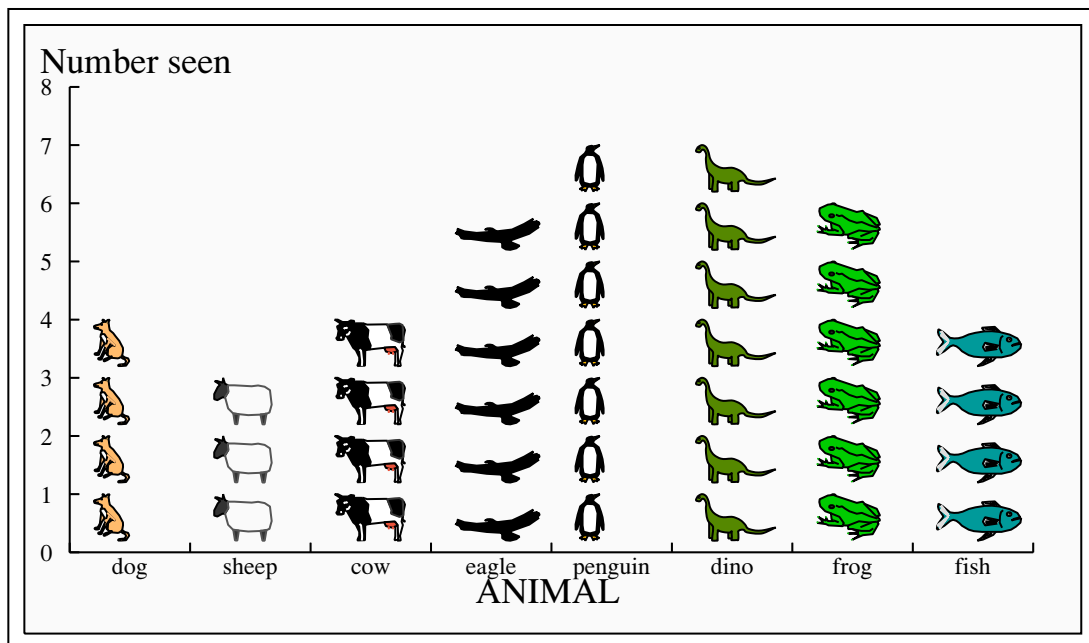


Figure 22.17

General graphs

General graphs are the same as those produced by **Utilities -> Simple Graph**. The system takes a block of data and attempts to draw a graph without further intervention, using one of the standard graph types

Techniques 23 - Decorating Graphs

You can edit a graph by making various changes and additions.

The graph editor has some similarity to a standard drawing package such as **!Draw**. Don't be misled; the editor is **not** a general-purpose drawing program but part of Schema2, and its features are designed accordingly.

Selecting Objects

A graph is made up from a collection of objects, which can include:

- an X-axis, which consist of lines, ticks, labels, etc.
- a Y-axis, which also consists of lines, ticks and labels
- a data picture, which includes histogram bars, points, and lines
- a legend box

As you edit a graph, you may add more objects including:

- lines
- rectangles
- circles
- ellipses
- segments of text
- copies of Draw files

When the mouse pointer is over the graph, click the **Select** button to select any object. The selected or **current** object is marked with a pale grey frame, and its type is shown in the title bar of the graph.

If two or more objects overlap, you can select them in rotation by clicking in the same place.

To de-select the current object, click **SELECT** on an empty part of the graph. The title bar will say: **Nothing Selected**

One useful state is to have the whole graph selected. You can do this with a menu entry, which is described below.

To summarise, you can select any one object in the graph. You can also have no object selected, and you can select the entire graph.

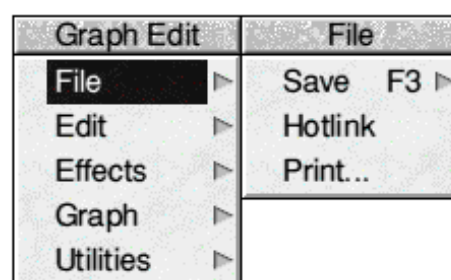
Dragging objects

In general you can move an object to any position you like. Click **SELECT** on the object and drag it to its new position. You cannot drag axes or data pictures, but you **can** move a legend box.

Graph Edit menu

If you click the **MENU** button over a graph, Schema2 shows the menu on the right.

File submenu - This menu lets you store or print graphs and is also shown on the right. It will be described in Section 24.



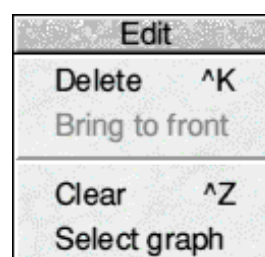
Edit submenu

Delete (Ctrl-K) - deletes the current object.

Bring to front - This ensures that the selected object is redrawn, making it appear in front of any other object. When the whole graph is redrawn the object stays at the front.

Clear (Ctrl-Z) - This entry clears the current selection so that no item is selected.

Select Graph - This menu item selects the whole graph.



The Effects Submenu

This submenu lets you choose effects to govern the appearance of objects in the graph. There is a group of entries for text objects and a group for lines. Fill colour applies to solid objects such as rectangles, circles and ellipses.

The meaning of the Effects menu depends on the currently selected item:

- If **no item is selected** then the system remembers any choice you make with the Effects menu and uses it as a default when you add new items. For example, if you choose 18-point red characters in Corpus Bold, then any text items you subsequently add will be in red 18-point Corpus Bold.
- If you have **an item selected**, then part of the Effects submenu is greyed out and only the usable entries are enabled. For example, if you select a rectangle then the Text effect options are disabled.

Any effect you choose with a selected object applies to that object only.

When you change the text effects on an axis or a legend box the graph is redrawn using the changed text style for that object.

- If the **whole graph** is selected, only the text effects are enabled. If you change a text effect, the entire graph is redrawn using the changed character style as a default.

Effects	
Text font	▶
Text size	▶
Text aspect	▶
Text colour	▶
Text background	▶
Fill colour	▶
Line width	▶
Line dashes	▶
Start cap	▶
End cap	▶
Cap length	▶
Cap width	▶
Line colour	▶

Next, we discuss the Effects menu in further detail:

Text font - lets you choose any font present in your machine.

Text size - covers any text size between 6-point and 72 point.

Text aspect - lets you set the ratio between the height and the width of your characters. Normal type has an aspect ratio of 1; thin characters have ratios of less than 1, whilst fat ones have aspect ratios which exceed 1.

Changing the aspect ratio is often a useful way to improve the appearance of axes.

Text colour - leads to a colour picker dialogue to choose the text colour.

Text background - is useful when characters are small enough to warrant anti-aliasing onto a coloured background.

Fill colour - lets you choose the internal colour of solid objects.

All the remaining options apply to lines. Line width and Line colour apply to solid objects, where they define the appearance of the border.

Line width - You can specify the line as one of:

- Thin
- 1
- 2
- 4
- Other

Line widths are given in 1/256 of an OS unit, so to have any visible impact the width should be some multiple of 256.

Thin lines are drawn as thinly as possible, and the values 1, 2 and 4 have special meanings for certain plotting devices. For high resolution output to laser printers or typesetters you should set the line to a minimum of 1.

Line dashes - A line can be drawn continuously, or as a series of dashes of various lengths. When you choose this item, you are offered a submenu with the entries

- None
- Dots
- Short
- Medium
- Long

Start cap and **End cap** - These let you choose how your lines start and stop. The difference between the options is only visible in thick lines. See the examples shown below

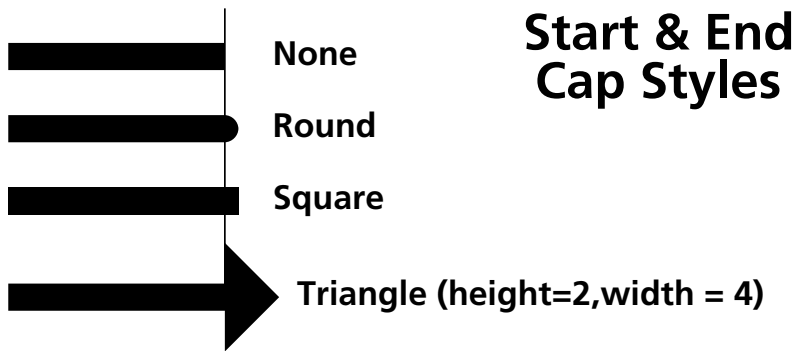


Figure 23.4

If you choose Triangle, you must specify the height and the width in terms of the thickness of the line itself. The height and width must be whole numbers between 0 and 15.

Line Colour - lets you choose the colour of the line.

The Graph Submenu

This menu is used to make changes to the whole graph. Various items may be disabled, depending on the type of graph and the selected object.

Three D effect - This option redraws the graph with a 3D effect. The menu entry is only enabled for histograms and pie-charts.

Grid - This menu item redraws the graph with the axis marks extended to a full rectangular grid, as shown below in Figure 23.6:

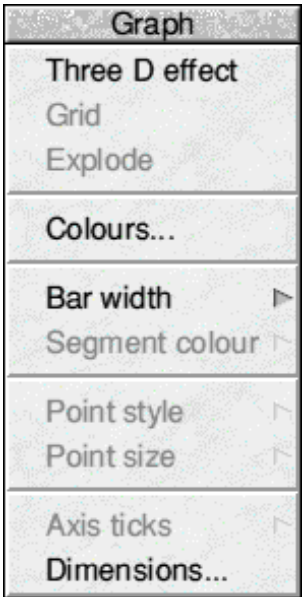


Figure 23.4

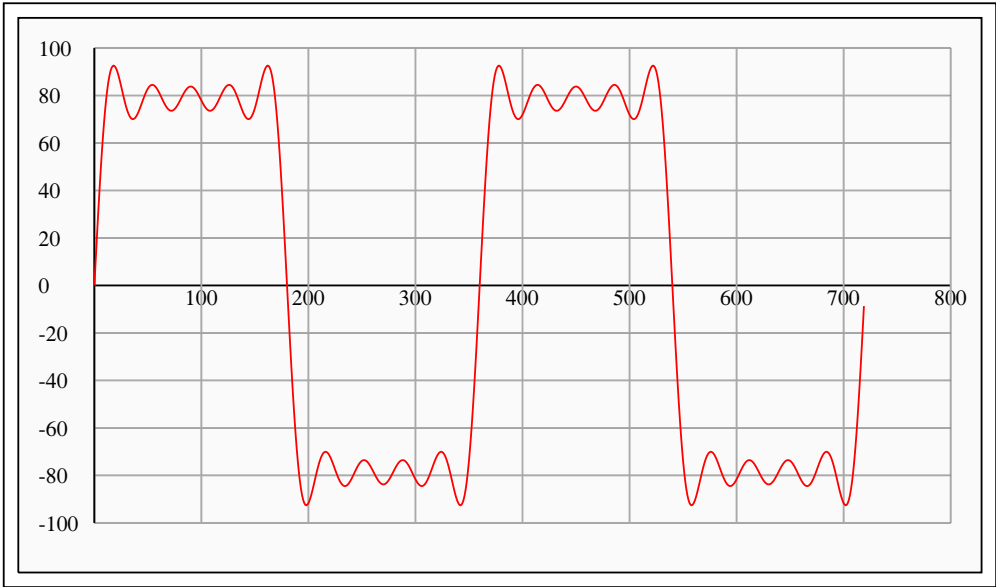


Figure 23.6

Explode - This entry is used to draw a pie chart with the segments spread out. An example is shown below in Figure 23.7. To move a slice of pie out, select it with a mouse click and choose **Explode** from the menu. A second click on **Explode** brings the segment in again.

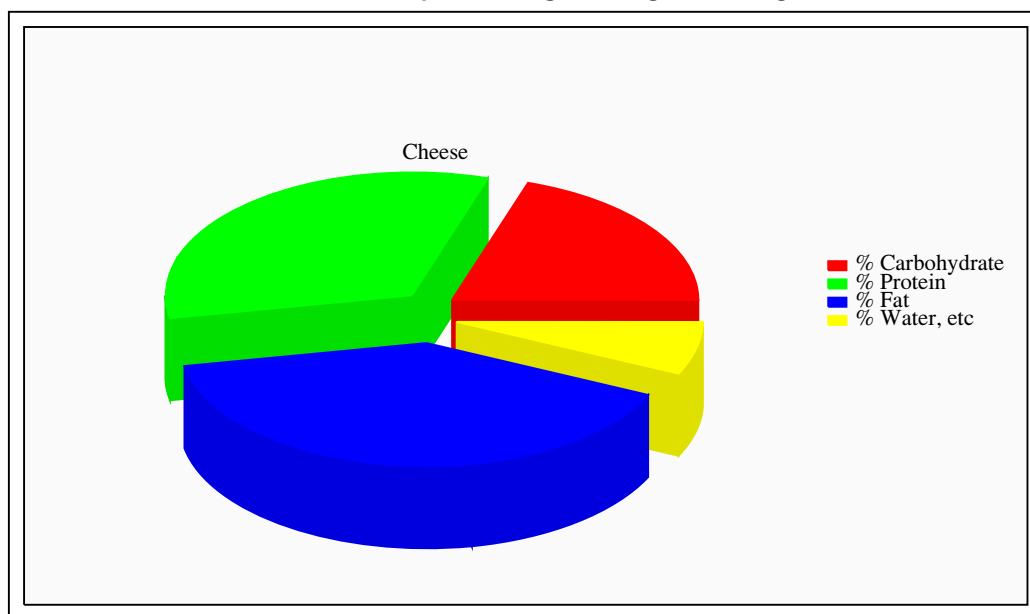


Figure 23.4

When you select a slice of pie the bounding box may not correspond exactly to the visible part of the segment as that part of the slice may be hidden by another slice and some large slices may have to be drawn as several pieces with invisible joins. The pie chart in Figure 23.8 illustrates this. If Schema2 tried to draw the large segment in one piece, either before or after drawing the small segment, the picture would not be correct.

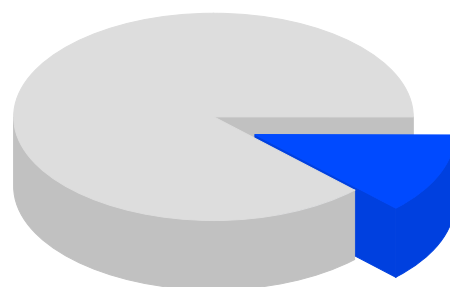


Figure 23.8

Colours - This entry leads to a dialogue which lets you change many of the colours used in the graph. The colour selection dialogue is shown to the right in Figure 23.9. In particular, you can modify:

- the background colour
- the border colour
- the colours used for each of the

data sets (up to five)

In the Graph colours dialogue the small squares show the currently selected colours. The larger panels are buttons which open up conventional colour picker dialogues. To open the colour picker click on the colour or the button.

Note that the default colours for graph plotting can be set from the Choices dialogue.

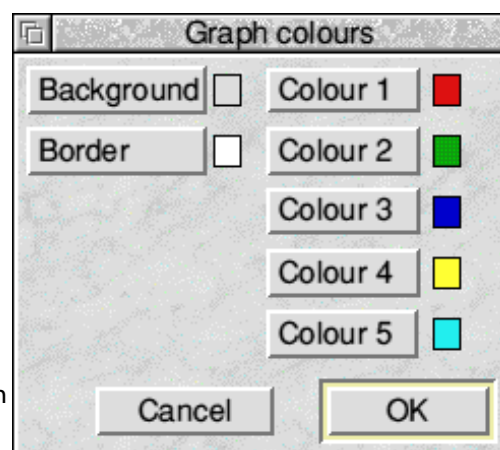


Figure 23.9

Segment colour - This option lets you choose the colour to be used for each segment in a pie chart. First select a slice; then bring up the **colour picker** dialogue by choosing Segment colour.

The colour you choose is used for the top of the visible segment. In a 3D display the sides of the slice are rendered in a slightly darker shade. In general the difference is only visible in a 256 colour mode.

Bar Width - This is only enabled for histograms. It sets the width of each bar as a fraction of the available width. The default width is 0.7

Point size and **Point style** are used to set the appearance of the markers in a Scatter diagram or Marked line plot. The system has five different styles of point marker:

X
+
s Triangle
u Diamond
n Square

When a Scatter diagram or Marked line plot is drawn, the marker types are allocated to the data sets in the given order, so that if there is only one data set the points appear as X's.

The **Point style** entry lets you choose any of the five basic shapes for the first data set, and the others are allocated cyclically.

The normal marker size is 12 points. This can be changed with the **Point size** entry.

With standard Schema2 graphs it is not possible to use different sized markers in the same graph.

Axis ticks - This is only enabled if you have selected an axis. It lets you decorate the axis with minor ticks in between the labelled marks.

If the axis is **linear** you can specify the number of minor ticks to go between each labelled mark, and the ticks will be evenly spaced.

If the axis is **logarithmic** you will get unequally spaced ticks at the values 2, 3, 4, 5, 6, 7, 8 and 9 regardless of the number you ask for.

Dimensions - This option leads to the dialogue shown in Figure 23.10:

This dialogue has three sections:

The **Overall size** gives the desired dimensions of the graph. The initial values are the same as the ones you gave in the Graph Options dialogue (see Section 21) but you can alter them and the graph will be redrawn accordingly.

The **Outer border** is an area which is not used by the graph plotting macros but remains free for additions such as logos or text. The width of the outer border can be adjusted on all four sides of the graph.

The **Inner border** is generally filled with the background colour but is not used for any other purpose. Again it can be used for decorations, etc.

The units of measurement are set in the Choices dialogue.

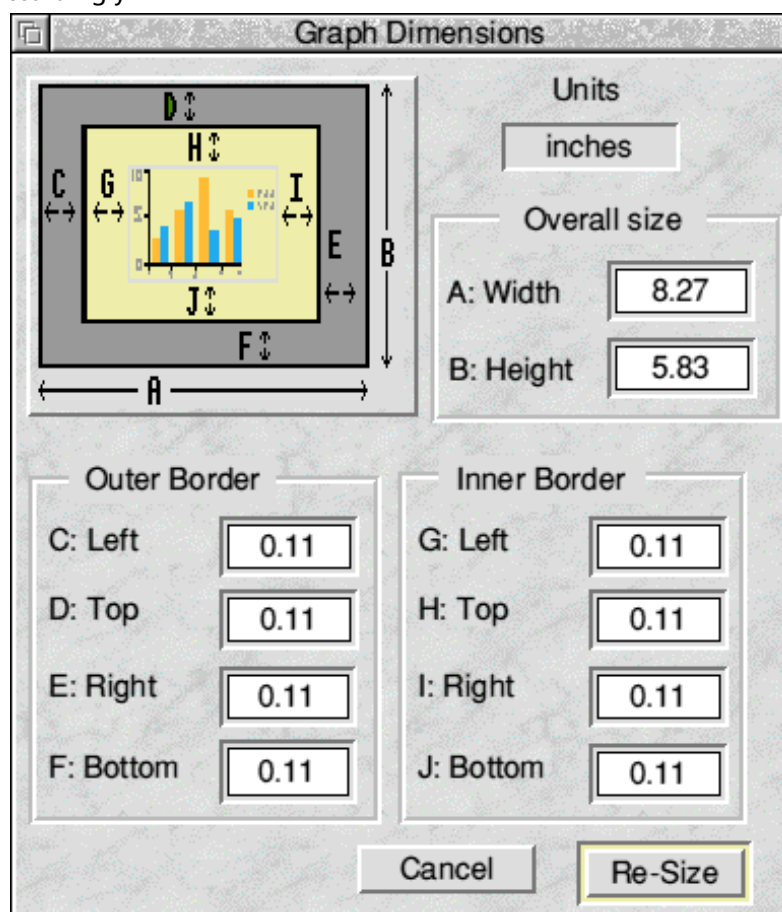


Figure 23.10

Utilities menu

This menu is used for several purposes. It has five entries, as shown on the right in Figure 23.11.

Scale view - This lets you see the graph at any scale between 10% and 999%.

Grid size and **Snap to grid** - These two entries control a 'snap grid' - a fine grid of lines which help you place and move objects in the graph.

If the Snap grid is on, the system adjusts the position of any object you move so that its upper left-hand corner lies at a grid crossing point. This makes it easy to get groups of items properly lined up.

Snap grid size controls the dimensions of the snap grid, and Snap to grid turns the feature on or off.

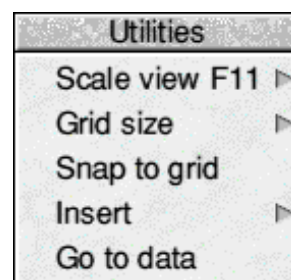


Figure 23.11

Go to data - This command reselects the data block used to draw the graph.

Insert - This entry lets you add new objects to the graph. It leads to a further submenu of object types shown on the right in Figure 23.12.

Insert submenu

Text - To insert text choose this item and click on the graph in the position of the first character. A character cursor appears and you can type the characters of the text. Correct errors with the **Delete** key. End the entry with **Return** or by clicking elsewhere on the graph.

To edit text which has been already inserted, select it and use the keyboard to delete and re-enter the text. You cannot use the cursor keys to move within the selected text.

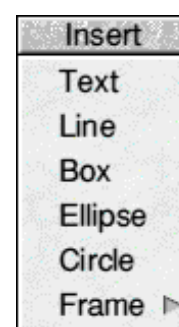


Figure 23.12

Line - To draw a straight line choose the Line item click with **SELECT** at one end of the line you want to draw and drag it to the other end of the line. You can use the Effects menu to vary the features of the line, such as its colour, width and type of ends.

Box - A box is a rectangle with its sides parallel to the borders of the graph. To draw a box select the Box item click with **SELECT** at one corner of the box and drag it to the opposite corner. The Effects menu lets you change the colour of the rectangle, and the colour and width of its border.

Ellipse - This option draws an ellipse with its axes parallel to the borders of the graph. Select the Ellipse item click with **SELECT** at one corner of the containing rectangle, and drag it to the opposite corner. You can change the colour and border of the ellipse with the Effects menu.

Circle - This option is the same as ellipse, except that the circle is drawn with a diameter equal to the width of the bounding box, and is placed at the centre of the box.

Frame - This lets you insert a Draw file, such as a company logo, anywhere in the graph by defining a box into which the file is dropped.

To define the bounding box select one of the options below and drag out a box before dropping the Draw file onto the graph.

- Full size - the picture is placed so that its lower left corner coincides with the lower left corner of the box. The size of the box is ignored.
- Scaled - the picture is placed with its lower left corner at the lower left of the box. However it is magnified or diminished so that one of its dimensions just matches the box, while the other is generally smaller. This preserves the aspect ratio of the picture.
- Squeezed - the picture is differentially magnified along each axis so that it fits the bounding box precisely. The aspect ratio is not generally preserved.

The graph shown in Figure 23.13 opposite contains various decorations. All four locomotives are copies of the same Drawfile:

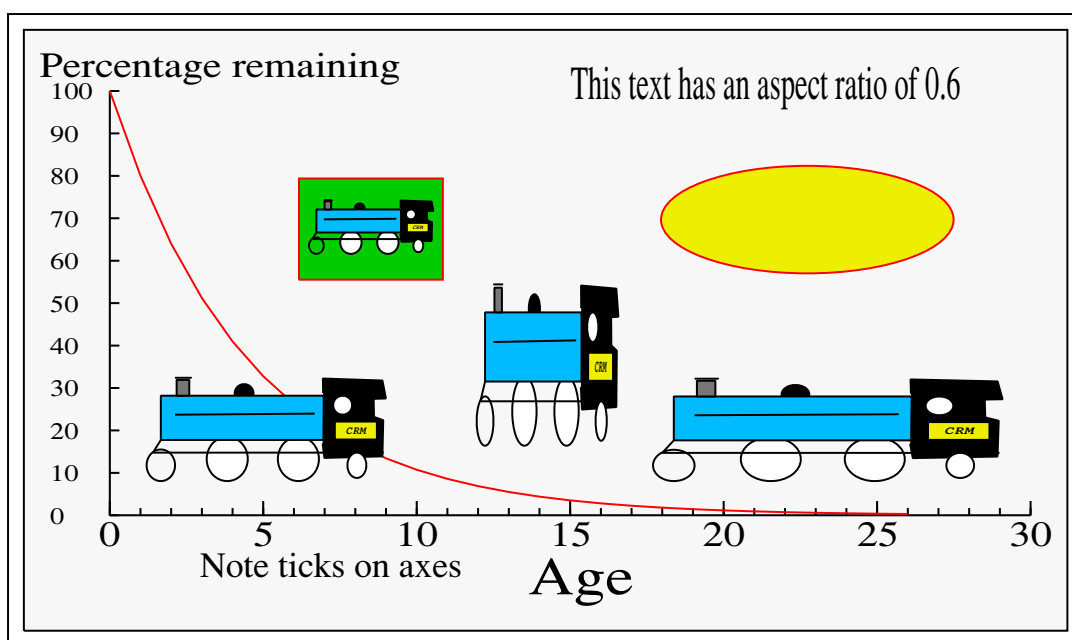


Figure 23.13

Techniques 24 - Storage of Graphs

Graphs displayed on the screen take up substantial amounts of memory. Each graph you draw must eventually be removed, otherwise the screen will get cluttered and the machine's memory may run short.

Discarding a graph

To **discard** a graph, click on the close icon in the top left hand corner of the window.

File menu

The File submenu lets you preserve the graph in various ways as alternatives to discarding it. The **File submenu** is accessed from the **Graph menu**, and is shown on the right in Figure 24.1:

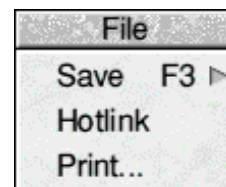


Figure 24.1

- Save -** This option saves the graph as a Drawfile. The entry leads to the standard Save dialogue where you can name the graph (if you wish) and drag the icon to any of the following:
- a disc directory.
 - another application, for example, a word processor to illustrate the text, or **!Draw** to further embellish the picture.
 - a cell in a Schema2 sheet, where the graph is handled just like any other Drawfile.

- Hotlink -** This facility lets you copy the graph back into a cell of the spreadsheet. Usually (but not necessarily) the cell will be a big cell.
- Provided that Automatic evaluation has not been disabled (see Section 25), a graph which has been hotlinked is redrawn every time any of its data points change. You can change your data and instantly see this reflected in your graph. However, evaluation can be slow, especially if there are many graphs in the spreadsheet or the graphs include numerous data points. Section 25 contains more on this subject.

To hotlink a graph into a spreadsheet:

1. Create the graph and do any decorating you want to do.
2. Select the cell where you want the graph to appear (this does not have to be in the same spreadsheet as the data and will often be a *big cell*). Ensure that the spreadsheet has the cream coloured **input focus** title bar.
3. Click **MENU** on the graph and choose **File -> Hotlink**.

When you plant a graph into a cell using a hotlink, Schema2 records a graph-producing formula in the cell. Whenever any of the variables in the formula changes, this formula is usually recalculated in the normal way and displays an updated graph in the cell.

Since graph calculation can be slow, you may wish to switch off automatic evaluation when you are entering data. The way to do this is described in Section 25.

- Print -** To print a graph, choose the **Print** entry in the submenu. This leads to the dialogue box shown on the right (Figure 24.2):

This dialogue lets you choose the number of copies, the magnification, and the orientation of the graph.

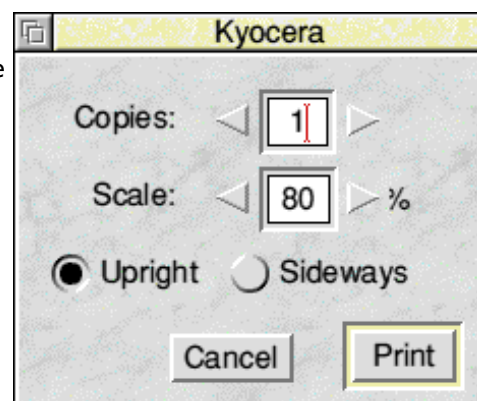


Figure 24.2

Sizing graphs

When you copy a graph into a cell, it is worth spending some effort to make it easily readable.

When you create a graph you will often know its intended destination and size. You can use the **Graph Options** dialogue, which is accessed from the main Schema2 menu via **Utilities -> Graphs**, to specify the graph size precisely. Either enter the dimensions by hand or select the target cell, usually a big cell and use **Fit to selection**.

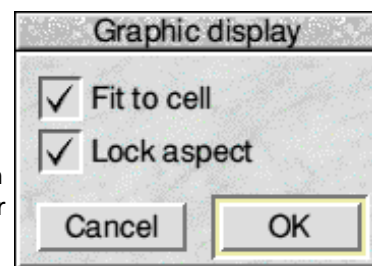


Figure 24.3

Alternatively you can use the **Effect -> Graphics** dialogue to auto scale (*Fit to cell*) the graphic to either a single cell or a Big cell. You can also choose to **Lock Aspect ratio** if you wish. The default when loading a graphic to a cell is to scale it to the cell whilst retaining the aspect ratio.

Re-opening a hotlinked graph

Sometimes you create a graph and hotlink it in to your spreadsheet. Later you decide that you want to add or change the decoration of the graph. To reopen a hotlinked graph for further editing:

1. Select the cell with the graph.
2. Click **MENU** over the spreadsheet and choose **Utilities -> Reopen graph** from the main menu.
3. Modify the graph using the **Graph Edit menu**.

If you want the re-styled graph to appear in the cell:

4. Select the cell with the graph again.
5. Click **MENU** over the graph and choose **File -> Hotlink** from the **Graph Edit menu**.

Techniques 25 - Evaluation

Evaluation is the process Schema2 uses to calculate the contents of cells and keep them up to date. Schema2 has several distinct evaluation modes, which control the time and the order in which individual cells are calculated.

You can classify evaluation modes in either of two ways:

1. Evaluation can be either **Automatic** or **Manual**. With automatic evaluation, Schema2 keeps the sheet up to date by doing all the necessary calculations as soon as you change a value in a cell; with manual evaluation, Schema2 defers computations until you specifically ask for them.
2. Evaluation can be **Intelligent** or **mechanical** - either By row or By column. Intelligent evaluation implies that after any change only the affected cells are re-evaluated, whereas mechanical evaluation means that after a change in any cell content, Schema2 works out **all** the cells in a fixed order.

Circular references

The concept of a circular reference helps in understanding intelligent evaluation.

If you enter formulae in to two cells, each of which refers to the other, you create a **circular reference**. An example of a circular reference is shown below in Figure 25.1:

	A	B
1	=B1+1	=A1+1

Figure 25.1

A key feature of a circular reference is that there is (in general) no way to fill in the values so that they all correspond to the formulae. In Figure 25.1, the numbers displayed in cells A1 and B1 cannot both tie up with the formulae in those cells. In mathematical terms, the pair of simultaneous equations -

$$\begin{aligned}A1 &= B1+1 \\ B1 &= A1+1\end{aligned}$$

have no solution.

Circular references are not limited to pairs of cells but can be caused by any ring of references of any length, including a single cell which refers to itself.

If you enter a formula which closes the ring and creates a circular reference Schema2 warns you but does not prevent it, since there are certain unusual applications where circular references are thought to be useful.

The very first spreadsheet programs offered only manual evaluation on a mechanical basis. With these sheets, you would usually have to evaluate the whole sheet several times to make all the cells converge to their correct values. To count evaluations, the users of these spreadsheets would often set up cells which referred to themselves, as shown in below in Figure 25.2.

	C
3	=C3+1

Figure 25.2

If the value of such a cell is initially zero, it will grow by 1 every time it is evaluated - providing a simple evaluation counter.

Schema2 is a modern spreadsheet, and if intelligent evaluation is used, the values in cells are guaranteed to be correct after each evaluation. There is therefore no need to count evaluations, and we vigorously deprecate the use of self-referring cells, or of circular references of any kind.

Automatic and Manual Evaluation

Automatic evaluation is the normal way of using a spreadsheet - you enter an item of data into any cell and the whole sheet (or set of sheets) is updated immediately.

Manual evaluation gives you a way to defer calculations which might be inconveniently long. For example, consider a sheet with some hotlinked graphs, where you want to change a whole group of numbers. With automatic evaluation, the graphs may be recalculated and redisplayed every time you alter a single cell. This could be a time-wasting process, since the intermediate forms of the graph, when only part of the data has been changed, are probably of no interest.

To avoid all this needless calculation, you can switch to the manual evaluation mode while you change all the data cells.

Intelligent and Mechanical evaluation

Intelligent evaluation minimises the amount of work needed to handle changes of values in cells. Two important characteristics of intelligent evaluation are:

- The only cells evaluated after a change are those which depend, directly or indirectly, on the changed value.
- The evaluation is done in an order which only requires each cell to be recalculated once. If a cell depends on another cell which needs to be re-computed, *that* cell is recalculated first provided there is no **circular reference**.

If the sheet contains any circular references, each cell affected by a change is calculated once, and the order of calculation is not defined.

Example

Consider the following model of a company's profitability, Figure 25.3, below.

	A	B	C	D	E
1	Results				
2		Turnover	=B8	Profit	=C2-B7*B4
3					
4	Cost Price	1.2			
5	Markup (percent)	45			
6	Sale price	=B4*(1+B5/100)			
7	Sales	10000			
8	Turnover	=B6*B7			

Figure 25.3

If the Sales (B7) are changed to 20000 then under **intelligent evaluation** cells will be evaluated in the following order:

B7 B8 C2 E2

No other cells depend on **B7**.

This sheet can be found as **Evaluation** in the **Examples** directory.

Evaluation by row and by column

Sometimes, to be compatible with certain primitive spreadsheets, you might want to evaluate mechanically by row rather than use intelligent evaluation. This means that the cells are worked out strictly in the order A1 B1 C1 ... A2 B2 C2 ... This process can generate incorrect results. If you evaluate repeatedly, the results will eventually converge to the correct answers given by intelligent evaluation.

You can also evaluate mechanically By column, in which case cells are worked out in the order A1 A2 A3 ... B1 B2 B3 ...

Example

Consider the spreadsheet **Evaluation** in the Examples directory.

Changing **Sales** (B7) will give the correct results if evaluation is **by column**, but the wrong results if evaluation is **by row** (cells C2 and E2 are evaluated before B8).

Evaluation **by row** and **by column** can be useful in **low memory** situations as it uses less memory than intelligent evaluation.

If evaluation is **by row** or **by column** then **sheet variables** are evaluated before the cells of the spreadsheet. The order of evaluating the sheet variables is undefined. If there are **multiple spreadsheets** to be evaluated then the order of evaluating the spreadsheets is undefined.

Evaluation mode dialogue

To display the Evaluation mode dialogue, shown below in Figure 25.4, choose **Utilities -> Evaluation -> Mode** from the main menu.

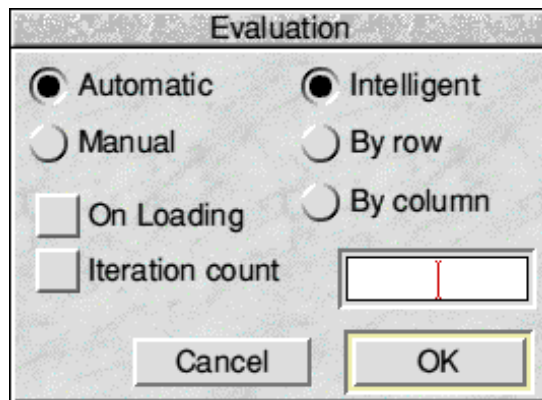


Figure 25.4

This lets you set the Evaluation mode for the spreadsheet.

It also lets you set an Iteration count. Setting an iteration count tells Schema2 to do the evaluation process an extra n times, where n is the number you have entered. This feature is mainly for compatibility with the original Schema.

Manual evaluation

If you select Manual evaluation mode evaluation of dependent cells is deferred until you give the command **Utilities -> Evaluation -> Do Do manual eval** from the main menu (keyboard short cut **Ctrl F10**).

Recalculate all

Selecting **Utilities -> Evaluation -> Recalculate all** from the main menu or using the keyboard short cut **Ctrl E** causes all the cells in all loaded spreadsheets to be recalculated. This is useful if recalculation failed previously for some reason or you want all the NOW() functions recalculated.

Values

By selecting **Utilities -> Evaluation -> Values** from the main menu you can control how some values are interpreted, via the **Values** dialogue shown below in Figure 25.5.

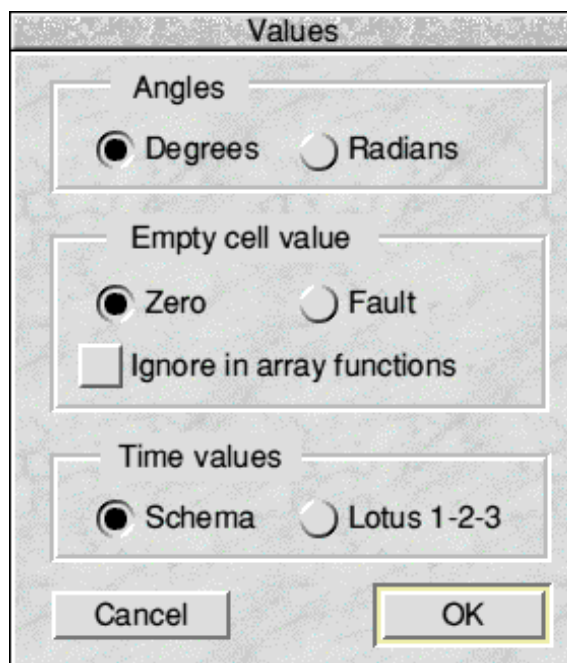


Figure 25.5

- Angles -** The Angles setting determines whether the trigonometric functions expect to work in terms of Degrees or Radians.
- Empty cell value -** The Empty cell value setting determines what happens when a formula refers to a cell that is empty. There are two options:
- **Zero** - the formula treats the empty cell as if it had the value 0
 - **Fault** - the formula generates an error **Reference to undefined cell**
- Time value -** The Time values setting determines the **time mode** used by the time functions and the number formatting. If your spreadsheet was originally a Schema or an Advance spreadsheet use the **Schema** time mode. If your spreadsheet was originally a **Lotus 1-2-3** spreadsheet or you think you might export it as a WK1 file use the **Lotus** time mode.
For an explanation of the differences see Section 17.

Final comment

The normal evaluation mode for spreadsheets is intelligent automatic. It is sometimes worth using the manual mode to speed up data entry, but you should not switch out of the 'intelligent' option unless you have some very good reason to do so.

Techniques 26 - Default documents and Choices

Schema2 lets you modify its operation in several ways to suit your particular circumstances, needs and preferences.

Default document

The default document is an empty spreadsheet used as a model or prototype for any new sheet you start. Although you should never need to access it directly it can be found inside the !Schema2 application directory with the name DefDoc.

The default document defines:

- number of rows and columns (unless you override it with the Create New Spreadsheet dialogue)
- **row** and **column** sizes
- appearance of **icon ribbon**, **labels**, **grid**, and the **scale**
- **print options**
- **styles**
- **evaluation mode**, **time mode**, **empty cell rule** and **angles mode**
- **sheet variables**

Changing the default document

To change the default document:

1. Create a new spreadsheet with the number of rows and columns you need. This spreadsheet forms the basis for your new default document.
2. Set up the attributes of the spreadsheet which you need for the default document.
3. Choose **Utilities -> Save as default** from the main menu.

The spreadsheet is saved as the new default document.

Choices dialogue

Some aspects of Schema2's operation apply over the whole application, and are not local to particular spreadsheets. These aspects can't be controlled by the **default document**. Instead, they are managed by the Choices dialogue. To bring up the Choices dialogue (Figure 26.1) choose Choices from the icon bar menu.

Units - You can choose measurement units for cell and sheet sizes. You have a choice of inches, centimetres, millimetres and points (1/72 of an inch each).

Default graph colours - Clicking **SELECT** on this button will bring up the Graph colours dialogue. The colours you choose are used for graphs when they are first created from the New graph dialogue.

Auto load - Schema2 allows you to specify whether some routines which are stored as **macros** should be loaded automatically when Schema2 is loaded. The advantage of loading them automatically is that it can prevent **Call of undefined macro** errors and requests for inserting discs on floppy disc only systems. The disadvantages are that they take up more memory and slow down the starting of Schema2. The routines are:

- **Graphics** - the standard graph types.
- **Schema1** - routines to implement some functions found in Schema1.
- **Lotus 1-2-3** - routines to implement some functions found in Lotus 1-2-3 WK1 files.

Discard sheet on close - If you set this option then Schema2 will discard your spreadsheet when you click on the close icon (as in !Edit). If you set this option off then it will keep your spreadsheet in memory but hidden (as in the original !Schema). See Section 6 for more details.

Black grid - This option controls the colour of the cell grid. If black is switched off, the grid is drawn in light grey.

Warn of cell contents overflow - If this is set then a warning will be issued if the contents of a cell become too large to fit in the space allocated.

Clear cell on key press - If this is set then when you type anything into a cell the old contents are automatically cleared to save you deleting them manually first.

Block Separator - This sets the character or characters used as Block Separators.

Cancel - If you click **SELECT** on **Cancel** then the dialogue will disappear without changing the Choices.

OK - If you click **SELECT** on **OK** then the settings you have set in the Choices dialogue will take effect immediately.

Save - If you click **SELECT** on **Save** then the settings you have set in the Choices dialogue will take effect immediately as if you had used OK, but also the settings are saved and used automatically by Schema2 on starting next time.

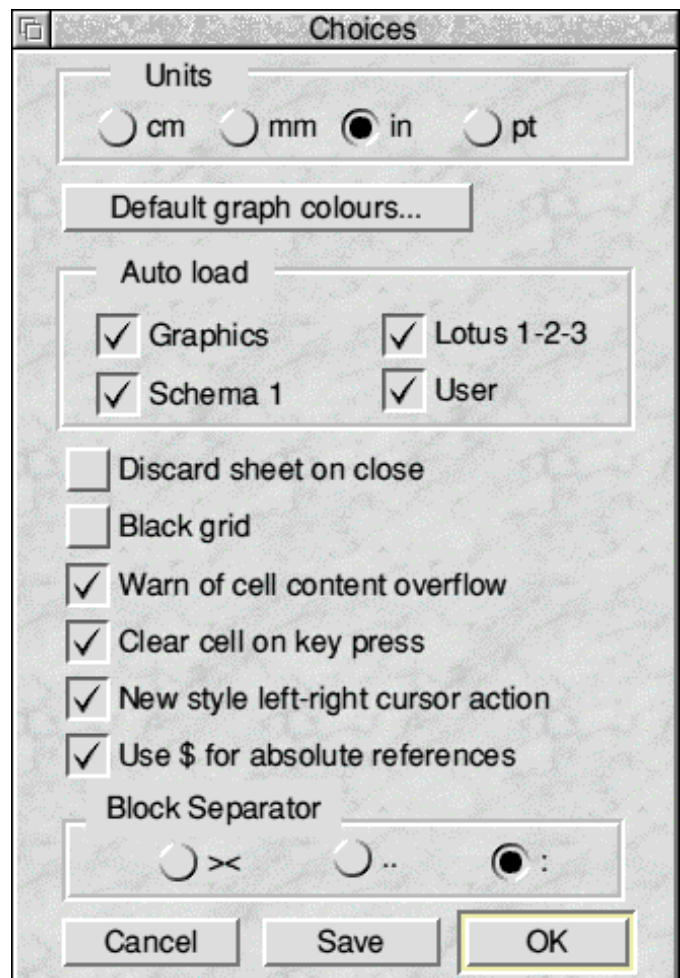


Figure 26.1

Techniques 27 - Resource files

Schema2 allows you to modify parts of the appearance and style of the user interface, by specifying **resource files** in conjunction with a **workspace**. You can modify:

- the **icon ribbon**
- the **menus**
- the **keyboard short cuts**
- the images used for **pictograms**

The resource files for the icon ribbon, menus and keyboard short cuts are kept in a directory called **Menus**. The resource files for the images used for pictograms are kept in a directory called **Pictures**.

You can modify Schema2 by editing the resource files. To do this successfully you must be competent in the following areas:

- opening directories, copying directories and files.
- using **!Edit** or another text editor on a text file to change the menus.
- using **!Paint** if you want to change or add to the icons on the icon bar.
- using **!Draw** if you want to change or add to the images used for pictograms.

Loading resource files with a workspace

When you load a **workspace** (see Section 13) Schema2 searches for a **Menus** sub directory in the same directory as the workspace. If there is a **Menus** sub directory then Schema2 reads the resource files from that directory, setting up new menus and icon ribbons. If there is no **Menus** sub directory, or any of the files are missing then Schema2 reads the missing resource file from its own **Menus** directory within the **!Schema2** directory. Similarly, if there is a **Pictures** directory it reads the list of pictogram images from the **Pictures** directory.

There is an example workspace, with resource files, in the **ResourceWS** sub-directory of the **Examples** directory. The resource files in this directory are the same as those used by Schema2 as default.

The directory structure is shown in Figure 27.1:

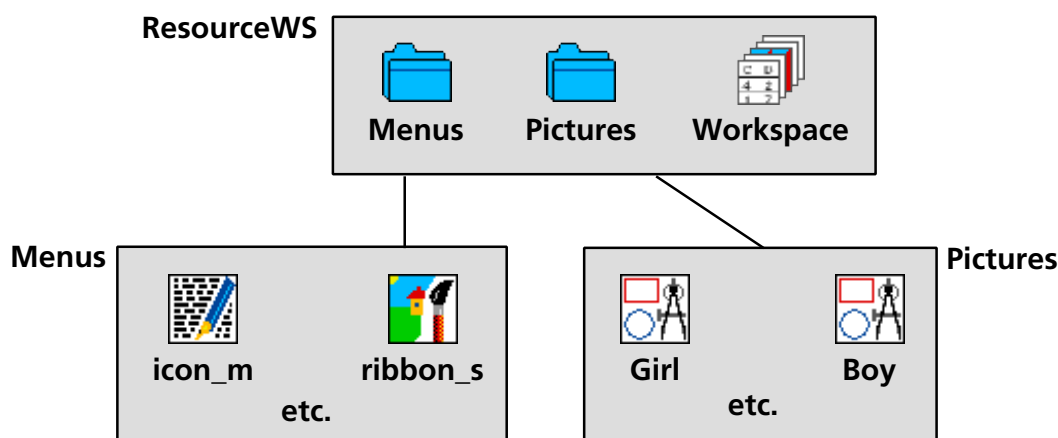


Figure 27.1

The simplest way to produce your own workspace with modified resource files is to copy this directory and edit the relevant files.

Menu Description Files

Schema2 has 5 distinct menus structures. Each one is described in its own file, as follows:

- | | |
|------------------------|-------------------------|
| • icon bar menu | Menus.icon_m |
| • main menu | Menus.window_m |
| • function pop-up menu | Menus.entry_m |
| • graphics menu | Menus.graphics_m |
| • style pop-up menu | Menus.style.m |

The menu structure description files are **text** files, and should be altered with **!Edit**.

To give an example, the menu description for the icon bar is quoted in full:

```
Record macro
Start,CAPTURE_MACRO
Stop>END_CAPTURE
Abort,ABORT_CAPTURE
.
Schema 2
Info> INFORM
Create> NEWSHEET
New view; NEWVMEN
Discard; DISCMEN
Record macro, RMAC
-
Save workspace>SWSP
Choices...,APPPREFS
Quit, QUIT
.
$
```

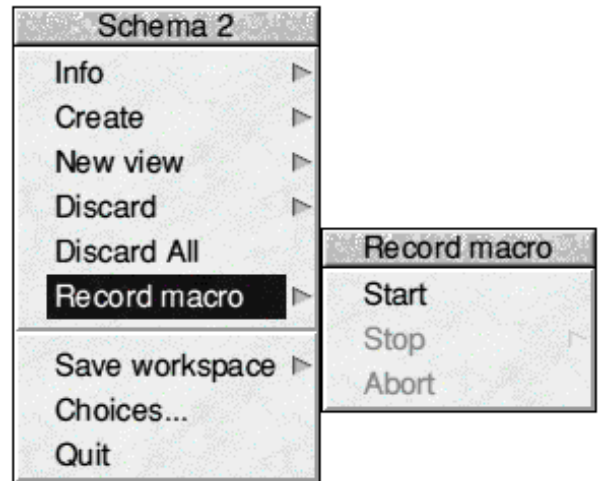


Figure 27.2

This file defines the familiar structure shown beside it in Figure 27.2.

The other menu description files are longer but are built on the same principles.

To modify a menu structure you need to understand the notation precisely. The details are:

- A menu description file contains one or more menus and is terminated by a \$ sign on a line by itself.
- Each menu starts with a title of not more than 12 characters and ends with a dot on a line by itself. When the menu is displayed the title appears at its head.

There is a rule about ordering: submenus must be described before their parent menus.

In our example there are two menus: **Record macro** (3 items) and **Schema2** (9 items). Record macro is a submenu of Schema2, so it is placed first.

The body of a menu consists of a number of entries, placed between the title and the terminating dot. Each entry must appear on a line by itself.

There are three kinds of entry:

1. A - (*minus sign*) indicates a line used to separate groups of entries.
2. A *submenu call* is the title of a submenu. Some submenu calls have *labels* attached to them. A label is a word, always spelled in capitals and preceded by a comma. In the case of submenu calls Schema2 uses these labels to tell it what messages to supply to **!Help**.

The **Record macro** entry in the Schema2 menu is an example. This submenu is called using the item: **Record macro, RMAC**

3. A *normal entry* is one which leads to an action, a dialogue, or a private submenu which is built by Schema2 every time it is used. For an example of a private submenu, take the New View item in the icon bar menu. This entry leads to a submenu which lists the sheets you have loaded. The submenu changes every time you load or discard a sheet, and you cannot set its contents in advance with a resource file.

The normal entry consists of three parts:

- a menu item - the sequence of characters, up to 40 in number, which appears in the menu.
- a separator - a comma, a semicolon, a > sign, etc.
- a label.

For example:

Create		>	NEWSHEET
<i>item</i>	<i>separator</i>	<i>label</i>	

The separator and label should be thought of as a pair. Each label always has the same separator.

Sometimes (although not in the icon bar menu) the item includes the description of a **keyboard short cut**. An example taken from **window_m** is

Cut ^X, CUTBLOCK

This defines **Ctrl X** to be the keyboard short cut for Cut as well as ensuring that the symbols **^X** appear on the menu.

Labels with Parameters

Certain labels are special in that they allow you to give them a parameter which tells Schema2 what item that menu operation should be performed on.

To set up a menu entry with a parameter enter the line in the menu description file as follows:

Menu item, LABEL Parameter

An example from the main menu (**window_m**) is:

Simple graph T, QUICKGR General

These labels and parameters should be left alone for the time being. Consult the Macro Manual to find out how to use them to link macros into Schema2.

Editing a Menu Description file

When you change the description of a menu structure, you may:

- change the order of items in a menu.
- modify the keyboard short cuts attached to the items.
- remove items altogether (including calls to submenus and the corresponding submenus themselves).
- add as many 'minus' items as you like.
- change the name of a menu or submenu, provided that you keep the length of the name to 12 characters and change all references to it accordingly.
- remove a submenu and place its items in the parent menu.
- collect a group of items from a menu and make them into a new submenu.
- change the item in a menu entry, provided that you keep the length to 40 characters at most.

You may **not**:

- invent any new items, except by using a label which expects a **parameter** (see below).
- change the separator or label in an item.

You can make any permitted change by simple editing. To give a simple example, here is an alternative version of the icon bar menu definition:

```

Capture
Give up, ABORT_CAPTURE
Start capture, CAPTURE_MACRO
Enough, END_CAPTURE
.
Spreadsheets
Discard; DISCMEN
Fresh view; NEWVMEN
.
My Schema
Information > INFORM
Make sheet > NEWSHEET
-
Capture
-
Spreadsheets
-
Save WSP >SWSP
Preferences..., APPREFS
Close down,QUIT
.
$

```

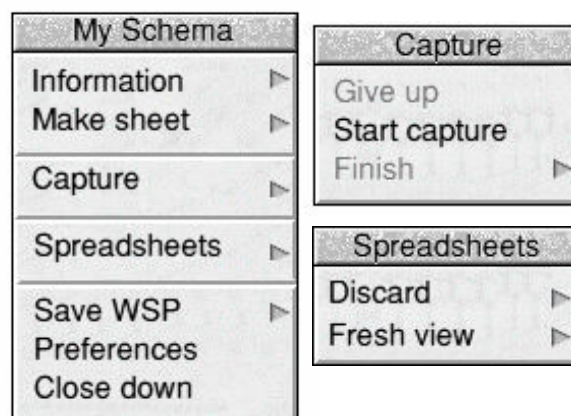


Figure 27.3

This defines the menu structure shown to the right in Figure 27.3.

If you use Schema2 with an invalid menu definition, it will stop loading the resource file and display an error message. There is no way to ignore such an error; you have to correct the file and try again.

Deleting Items

It is easy to delete an item from a menu description, but you should remember that every time you do so, you reduce the functionality of !Schema2. For example, if you remove the Cell borders entry there will be no way to put borders into cells, and no way to remove them from imported sheets.

Some deletions, such as the Save submenu, will make the system demonstrable but unusable, except to someone who knows how to put the deleted features back.

Perhaps the most sensible items to remove are function names which the users may not have met. You can protect them from items such as hyperbolic and financial functions.

Changing keyboard short cuts

Here you must observe certain restrictions:

1. You can only use a limited set of keys as short cuts:
 - The function keys **F1** to **F11** (but not **F12**)
 - The function keys **F1** to **F12**, together with **Control**. When you type the menu description file refer to these keys with Control as **^F1** to **^F12**.
 - Letters A to Z, but excluding M and U, together with the Control key. You write these as:
A B ... Z
2. The same short cut must not be used more than once in the same menu structure.
3. If a menu includes several short cuts, use extra spaces to make sure they are lined up if you want your menu to look right. For example:

Alignment	
Left	F5,EFLEFT
Centre	F6,EFCENTRE
Right	F7,EFRIGHT
Justify	F8,EFDDOUBLE

4. Certain keyboard short cuts are accepted as standards for ACORN software. For example, **F3** always means Save. If you reassign this key - perhaps to Delete - you risk becoming unpopular with other users.

Once you have changed the short cuts in the menu description, the new key assignments are used automatically. No further changes are needed.

The Icon Ribbon

You can define icons in the **icon ribbon** and set their meaning. The icon ribbon uses two resource files, an icon ribbon definition file called **Menu.ribbon_m** and a sprite file defining the sprites used for the icons, called **Menus.ribbon_s**.

The sprite file should also contain the sprites for the **tick**, **cross**, **sum**, **equals**, **pointer** and **function** icons.

The sprite file actually comes in three different varieties, which are used according to the current screen mode. They are called

Menus.ribbon_s22	(Used for high-resolution colour modes such as Modes 20,27,28, or 31)
Menus.ribbon_s23	(Used for mode 23: high-resolution monochrome)
Menus.ribbon_s	(Used for all other modes)

Each icon file is of type **Sprite**, which implies that it holds a collection of sprites. In **ribbon_s22** and **ribbon_s23** the icons are 22 pixels high and 22 pixels wide. In **ribbon_s** they are 11 pixels high and 22 pixels wide.

In principle, every icon you use in the icon ribbon should have a named sprite in *each* of these three files. In practice, you don't need to include a sprite in an icon file if you have no intention of using that group of screen modes.

The icon ribbon definition file serves to link icons to specific menu labels. The structure is similar to menu definition files. The standard version is:

```
print,PRINT
graph1,QUICKGR General
graph2,NEWGR
-
bold, EFBOLD
italic, EFITALIC
format, EFFORMAT
-
cut, CUTBLOCK
copy, CBL
paste, PASTEBLOCK
.
$
```

Each line links an icon to a menu entry. For example, the line

bold, EFBOLD

ensures that:

- the bold icon appears on the icon ribbon.
- a click on this icon has the same effect as choosing the menu entry labelled **EFBOLD**.

The menu labels are the same as those used in the main menu (**window_m**). You will need to look these up to ensure that you attach icons correctly.

The - (minus) entries serve to space the icons into groups.

The list of entries is terminated by a . and a \$ sign, each on a line by itself.

Altering the icon ribbon

There are various ways you can alter the special icon bar:

To alter the appearance of an icon: Use **!Paint** to edit the icon in the resource file. Don't change its name or its size.

To remove an icon from the bar: Use **!Edit** to take out its entry in resources.ribbon_m.

To change the order of the icons: Use **!Edit** to change the order of the entries in resources.ribbon_m.

To add a new icon

1. Design a new icon, choose a name, and place a copy in each of the resource files. The size **must** be correct.
2. Decide which menu entry the icon is to duplicate, and note the label for this entry. You can only set up icons for operations in the main menu.
3. Use **!Edit** to plant a new item in ribbon_m. The new items consists of the icon name, a comma, and the label.

For example, you decide to add a new icon for reopening graphs:

- 1a. Using mode 27, you design a 22*22 icon called **reopen** and add it to file **Menus.ribbon_s22**.
- 1b. Using mode 23, you design another 22*22 icon called **reopen** and add it to file **Menus.ribbon_s23**.
- 1c. Using mode 12, you design a third icon called **reopen**. (this time of size 11*22) and add it to file **Menus.ribbon_s**.
2. Discover from **windows_m** that the label for Reopen graph is **REEDGR**.
3. Use **!Edit** to add the line

reopen, REEDGR to **Menus.ribbon_m**.

The Pictures directory

The **Pictures** directory holds about 50 small drawfiles which produce the images in **pictograms**. You can change or remove existing Draw files, and add new ones up to a limit of 77 in the directory. Be careful before you remove a Draw file, since the system may then be unable to redraw an existing graph.

When you change a file or add a new one remember that very large Draw files will slow down plotting and use up a lot of memory. The files in the standard CRM set are carefully designed to be as simple as possible, and they have an average length of about 2000 bytes each. The elephant uses only 1808 bytes.

Modifying the resource files in !Schema2

You can modify the resource files inside **!Schema2** in the **Menus** and **Pictures** directories. This will affect all users of Schema2 unless they have resource files attached to their workspaces. Therefore great care should be taken when doing this - **make a backup!**

Appendix A

Definition of Schema2 Standard Functions

Part 1 - Overview

1. Introduction

General rules about the use of standard functions are given in Module 2 -**Techniques 20**.

1.1 Function arguments

Each standard function expects exactly the right number of arguments. Each function also needs the right type of argument(s), but when necessary the arguments are converted automatically. Whole numbers, decimal fractions and strings can all be converted from one to another; only blocks cannot be changed into any other type. A function call is rejected if there is confusion between blocks and other types of parameter.

1.2 Function and macro definitions

Each of the following chapters defines a group of functions or macros. The groups are:

Algebraic functions:	Part 2
Scientific functions:	Part 3
Trigonometry functions:	Part 4
String functions:	Part 5
Array functions:	Part 6
Temporal functions:	Part 7
Random functions:	Part 8
Statistical functions:	Part 9
Financial functions:	Part 10
Matrix functions:	Part 11
Miscellaneous functions	Part 12
Lookup functions:	Part 13
Database macros:	Part 14
Useful macros:	Part 15
Lotus 1-2-3 macros:	Part 16

Schema2 also supports a number of functions which can only be invoked from macros. These functions are described in the Macro Manual.

1.3 Lotus 1-2-3 Compatibility macros

To use the macros described in parts 14,15 and 16 you need to ensure that the macros are loaded. To do this ensure that **Lotus 1-2-3** is set for Auto load in the Choices dialogue.

Note: In this section we have consistently used upper case names for functions. The equivalent lower case or mixed case names work equally well.

Part 2 - Algebraic Functions

2.1	ABS Syntax: Result:	Absolute value of a number ABS (number) The absolute value of the argument- that is, the value of the number without its sign.
	Examples:	ABS(6.9) = 6.9 ABS(-7.5) = 7.5
2.2	CEIL Syntax: Result:	Smallest integer not less than a number CEIL (number) Smallest whole number equal to or greater than the argument.
	Examples:	CEIL (4.5) = 5 CEIL(77) = 77 CEIL (-6.5) = -6
2.3	FLOOR Syntax: Result:	Largest integer not more than a number FLOOR (number) Largest whole number equal to or less than the argument.
	Examples:	FLOOR (4.5) = 4 FLOOR(77) = 77 FLOOR (-6.5) = -7
	Comments:	Equivalent to INT
2.4	INT Syntax: Result:	Largest integer not more than a number INT (number) Largest whole number equal to or less than the argument.
	Examples:	INT (4.5) = 4 INT(77) = 77 INT (-6.5) = -7
	Comments:	Equivalent to FLOOR
2.5	ROUND Syntax: Result:	Rounds a number ROUND (number, integer) The value of the first argument, rounded to the number of decimal places specified by the second argument.
	Examples:	ROUND(4.73,0) = 5 . ROUND(4.73,1) = 4.7 ROUND (-5.69,1) = -5.7 ROUND (4.5,4) = 4.5000
2.6	SIGN Syntax: Result:	The sign of a number SIGN (number) -1 if the argument is negative 0 if the argument is zero +1 if the argument is positive.
	Examples:	SIGN (-34.8) = -1 SIGN (G5 - G5) = 0 SIGN (0.000001) = 1

Part 3 - Scientific Functions

3.1	COSH Syntax: Result:	Hyperbolic cosine of a number COSH (number) The hyperbolic cosine of the argument.
3.2	EXP Syntax: Result: Example:	Exponential EXP (number) e raised to the power of the number. EXP(1) = 2.718...
3.3	LN Syntax: Result: Comments:	Natural logarithm LN (number) The natural logarithm of the number (to the base e). This function generates a fault if the argument is not greater than 0.
3.4	LOG Syntax: Result: Comments:	Logarithm to base 10 LOG (number) The logarithm of the number to the base 10. This function generates a fault if the argument is not greater than 0.
3.5	SINH Syntax: Result:	Hyperbolic sine of a number SINH (number) The hyperbolic sine of the argument.
3.6	SQRT Syntax: Result: Comments:	Square root of a number SQRT (number) The square root of the argument. Generates a fault if the argument is less than 0.
3.7	TANH Syntax: Result:	Hyperbolic tangent of a number TANH (number) The hyperbolic tangent of the argument.

Part 4 - Trigonometrical Functions

4.1	ACOS Syntax: Result: Comments:	Arc cosine of a number ACOS (number) The inverse cosine of the argument. The argument is the cosine of an angle. The result is delivered in radians if the current mode is in radians; otherwise in degrees. The function generates a fault if the argument is outside the range -1 to +1.
	Example:	$\text{ACOS}(0) = 90 \quad (\text{Degrees})$ $= 1.5707... \quad (\text{Radians})$
4.2	ACOT Syntax: Result: Comments:	Arc cotangent of a number ACOT (number) The inverse cotangent of the argument. The argument is the cotangent of an angle. The result is delivered in radians if the current mode is in radians; otherwise in degrees.
4.3	ASIN Syntax: Result: Comments:	Arc sine of a number ASIN (number) The inverse sine of the argument. The argument is the sine of an angle. The result is delivered in radians if the current mode is in radians; otherwise in degrees. The function generates a fault if the argument is outside the range -1 to +1.
4.4	ATAN Syntax: Result: Comments:	Arc tangent of a number ATAN (number) The inverse tangent of the argument. The argument is the tangent of an angle. The result is delivered in radians if the current mode is in radians; otherwise in degrees.
4.5	ATAN2 Syntax: Result: Comments: Examples:	Arc tangent of a ratio ATAN2(x, y) x and y are numbers. The function returns the inverse tangent of y/x. If x is 0 and y is not 0 the function returns +90 or -90 (+pi/2 or -pi/2) depending on the sign of y. The quadrant of the returned angle depends on the signs of x and y. If x=0 and y=0 a fault is returned. $\text{ATAN2}(0,1) = 90 \quad (\text{Degrees})$ $= 1.5707... \quad (\text{Radians})$ $\text{ATAN2}(-1,-1) = 135 \quad (\text{Degrees})$ $= -2.3561... \quad (\text{Radians})$
4.6	COS Syntax: Result: Comments:	Cosine of an angle COS (number) The cosine of the argument. Argument is taken as radians if current mode is radians; otherwise degrees.
4.7	COT Syntax: Result: Comments:	Cotangent of an angle COT (number) The cotangent of the argument. Argument is taken as radians if current mode is radians; otherwise degrees.
4.8	DEG Syntax: Result: Example:	Convert radians to degrees DEG (number) The angle in degrees. $\text{DEG}(3.14159265358979 / 2) = 90$
4.9	RAD Syntax:	Convert degrees to radians RAD (number)

	Result:	The angle in radians
	Example:	RAD(45)= 0.78539....
4.10	SIN	Sine of an angle
	Syntax:	SIN (number)
	Result:	The sine of the argument.
	Comments:	Argument is taken as radians if current mode is radians, otherwise degrees.
4.11	TAN	Tangent of an angle
	Syntax:	TAN (number)
	Result:	The tangent of the argument.
	Comments:	Argument is taken as radians if current mode is radians; otherwise degrees. Note that TAN(90) should be infinite but in Schema2 it evaluates to a very large number.

Part 5 - String Functions

Note: The 'index' of a character in a string is its ordinal position in the string. The first character has index 1.

- | | | |
|-----|--|--|
| 5.1 | ASC
Syntax:
Result: | ASCII code of first character in a string
ASC (String)
The ASCII code of the first character in the string. |
| | Example: | ASC("Help") = 72 |
| 5.2 | CHR
Syntax:
Result: | Convert number into character
CHR (number)
A string with a single character, whose ASCII code is the argument. |
| | Example: | CHR(72) = "H" |
| 5.3 | FIND
Syntax:
Result: | Find the first occurrence of a string in another
FIND (string1, string2)
If string2 occurs in string1 , the function returns the index of the first character in the first occurrence. If the string does not occur the function returns -1. |
| | Examples: | FIND("Transylvania", "syl") = 5
FIND("Lola Sapola", "la") = 3
FIND("Schema2", "123") = -1 |
| 5.4 | LEFT
Syntax:
Result: | Left side of a string
LEFT (string, integer)
A string which contains the given number of characters taken from the left of the argument string, or the whole of the argument string, whichever is shorter. A count of zero or less gives an empty string. |
| | Examples: | LEFT("Archimedes", 4) = "Arch"
LEFT("Archimedes", 50) = "Archimedes"
LEFT("Archimedes", 0) = "" |
| 5.5 | LEN
Syntax:
Result: | Number of characters in a string
LEN (string)
The number of characters in the string. |
| | Examples: | LEN("Limpopo") = 7
LEN (" ") = 0 |
| 5.6 | LOWER
Syntax:
Result: | Change upper case characters to lower case
LOWER(string)
A copy of the argument string with all upper case characters changed to lower case. |
| | Example: | LOWER ("King George IV") = "king george iv" |
| 5.7 | MID
Syntax:
Result: | Extract characters from a string
MID (string, number1, number2)
Normally, a substring of the argument string, starting at character p (number1) and end ending at character q (number2). However, there are some special cases:

If p > q , their values are interchanged.
If p <= 0 , the result starts at the first character of the argument.
If q is greater than the length of the string, the result stops at the end of the string.
If p = q = 0 , the result is an empty string. |

Examples: MID("Archimedes",3,7) = "chime"
 MID("Archimedes",3,3) = "c"
 MID("Archimedes",8,4) = "himed"
 MID("Archimedes",7,30) = "edes"
 MID("Archimedes",4,0) = "Arch"
 MID("Archimedes",0,0) = ""

5.8

RIGHT**Right side of a string****Syntax:**

RIGHT (string, integer)

Result:

A string which contains the given number of characters taken from the right of the argument string, or the whole of the argument string, whichever is shorter. A count of zero or less gives an empty string.

Examples:

RIGHT("Archimedes",4) = "edes"
RIGHT("Archimedes",50) = "Archimedes"
RIGHT("Archimedes",0) = ""

5.9

UPPER**Change lower case characters to upper case****Syntax:**

UPPER(string)

Result:

A copy of the argument string with all lower case characters changed to upper case.

Example:

UPPER ("King George IV") = "KING GEORGE IV"

Part 6 - Array functions

Note: All the functions in this group take a block or an array as their first (or only) argument. Unless it is stated to the contrary, empty cells are treated as follows:

If the Empty cell value rule is 'fault', an empty cell in the argument block generates the error **Reference to undefined cell**. Otherwise it is taken as zero. Note that this may sometimes lead to unexpected results, for instance, if you AVERAGE() an area with blank cells.

All the examples quoted in this section refer to this spreadsheet (**AgeandIQ** in the **Examples** directory).

	A	B	C	D	E	F	G	H	I	J	K
1	Name	Jim	Harriet	Carl	Rachel	Clare	Fred	Tony	Ian	Gavin	Helen
2	Age	36	34	38	32	38	33	36	34	37	35
3	Height	190	178	172	185	175	190	180	178	193	156
4	IQ	104	73	80	109	100	95	100	114	98	109

- 6.1 ALL** **Check a block for any zero values**
Syntax: ALL(block)
Result: +1 if all the values in the block are non-zero. Otherwise 0.
Example: ALL(B2...K2)=1
- 6.2 AVERAGE** **Average of the entries in a block**
Syntax: AVERAGE(block)
Result: The arithmetic mean of the entries in a block.
Example: AVERAGE(B2...K2)=35.3
- 6.3 COUNT** **Number of non-zero entries in a block**
Syntax: COUNT(block)
Result: The number of entries in the block which are non-zero.
Comment: Empty cells in the block always count as zero, unless the **empty cell value** is **fault**.
Example: COUNT(B2...K2) = 10
- 6.4 ESTDEV** **Estimated standard deviation of a block**
Syntax: ESTDEV (block)
Result: The best estimate of the standard deviation of the parent population from which the values are taken.
Comment: ESTDEV (block) gives a value greater than STDEV(block) by a factor of $\sqrt{n/(n-1)}$, where n is the number of entries in the block.
Example: ESTDEV(B2...K2) = 2.05
The values in row B are actually calculated using a gaussian of standard deviation 2.
- 6.5 MAX** **Largest number in a block**
Syntax: MAX (block)
Result: The largest number in a block.
Example: MAX(B4...K4) = 114
The highest IQ (Ian's) is 114.
- 6.6 MIN** **Smallest number in a block**
Syntax: MIN (block)
Result: The smallest number in a block.

6.7	PRODUCT Syntax: Result:	Product of entries in a block Product(block) The product of all the entries in the block.
6.8	STDEV Syntax: Result: Comments:	Standard deviation of entries in a block STDEV(block) The square root of the variance of the elements in the block. See ESTDEV (6..4 above).
6.9	SUM Syntax: Result:	Sum of all the elements in a block SUM(block) The sum of all the elements in the block.

Part 7 - Time functions

Note: See Techniques Section 17 for an explanation of **temporal values** and **time modes**.

All calculations with times are based on the useful but false assumption that the Gregorian Calendar extends indefinitely into the past. In fact the Gregorian Calendar was introduced to Britain in the 18th Century and in various countries at different times up to the beginning of the 20th Century.

The Gregorian Calendar would have a year zero but historians assume that 1 AD followed directly after 1 BC. One departure from the Gregorian Calendar is that Schema2 ignores leap seconds like the one between 1988 and 1989.

All the temporal functions use temporal values, and carry out various conversions between temporals and calendar dates and times. They work in either Schema or Lotus time mode.

All the examples use a temporal value of 62904514302 (**Schema time mode**), which is 12.11:42 on Monday July 12th 1993. (The equivalent temporal value in **Lotus time mode** would be 34162.508125).

7.1	DAYNUMBER Syntax: Result:	Find day of the week DAYNUMBER(temporal) The day of the week represented by the temporal, coded as Sunday = 0 .. Saturday = 6.
	Example:	DAYNUMBER(62904514302) = 1
7.2	DAYS Syntax: Result:	Find day of the month DAYS(temporal) The day of the month as an integer in the range 1-31.
	Example:	DAYS(62904514302) = 12
7.3	HOURS Syntax: Result:	Find hour of the day HOURS(temporal) The hour of the day as an integer in the range 0-23.
	Example:	HOURS(62904514302) = 12
7.4	MINUTES Syntax: Result:	Find minute of the hour MINUTES(temporal) The minute of the hour as an integer in the range 0-59.
	Example:	MINUTES(62904514302) = 11
7.5	MONTHS Syntax: Result: Example:	Find month of the year MONTHS(temporal) The month of the year as an integer in the range 1-12. MONTHS(62904514302) = 7
7.6	MONTHSADD Syntax: Result:	Advance n months MONTHSADD(temporal, integer number) The second argument should be an integer number, n , which specifies how many months to add. n may have a negative value. This function generates a temporal value exactly n months later than the given temporal. When a date at the end of long month is advanced to a shorter month, the last day of the new month is used. For example, MONTHSADD applied to January 31st, with n equal to 1, would give February 28th, or February 29th in a leap year.
	Examples:	MONTHS(MONTHSADD(62904514302, 1)) = 8

		Going backwards by three months: <code>MONTHS(MONTHSADD(62904514302, -3)) = 4</code>
	Note:	The function <code>MONTHADD(temporal)</code> is equivalent to <code>MONTHSADD(temporal, 1)</code> . It is for backward compatibility with Schema only.
7.7	NOW	Find the time now
	Syntax:	<code>NOW()</code> (No arguments)
	Result:	A temporal which gives the time of the function call.
	Example:	<code>NOW() = 62904514302</code>
7.8	SECONDS	Find seconds of the minute
	Syntax:	<code>SECONDS(temporal)</code>
	Result:	The second of the minute as an integer in the range 0-59.
	Example:	<code>SECONDS(62904514302) = 42</code>
7.9	TIME	Find equivalent of a temporal
	Syntax:	<code>TIME(temporal)</code>
	Result:	A string which states the date and time represented by the temporal argument in the form <code>yyyy/mm/dd/hh/mm/ss</code>
	Example:	<code>TIME(62904514302) = "1993/7/12/12/11/42"</code>
7.10	TTIME	Convert standard time to temporal value
	Syntax:	<code>TTIME(year, month, day, hour, minute, second)</code>
	Result:	This function takes six integer arguments which represent the year, month(1-12), day (1-31), hour(0-23), minute(0-59) and second (0-59) respectively, and generates a temporal value for that time. If the arguments are out of range a fault is generated.
	Example:	<code>TTIME(1993,7,12,12,11,42) = 62904514302</code> In Lotus time mode: <code>TTIME(1993,7,12,12,11,42) = 34162.508125</code>
7.11	YEARS	Find year of a date
	Syntax:	<code>YEARS(temporal)</code>
	Result:	The year of the date as a signed integer.
	Example:	<code>YEARS(62904514302) = 1993</code>
7.12	YEARSADD	Advance n years
	Syntax:	<code>YEARSADD(temporal, integer number)</code>
	Result:	The second argument should be an integer number, n , which specifies how many years to add. n may have a negative value. This function generates a temporal value exactly n years later than the given temporal. When applied to February 29th, the result is February 28th. if the new year is not a Leap year.
	Examples:	<code>YEARS(YEARSADD(62904514302, 1)) = 1994</code> Going backwards by four years: <code>YEARS(YEARSADD(62904514302, -4)) = 1989</code>
	Note:	The function <code>YEARADD(temporal)</code> is equivalent to <code>YEARSADD(temporal, 1)</code> . It is included for backward compatibility only.

Part 8 - Random functions

Note: These are pseudo-random functions which return random numbers with particular distributions.

The seed for the pseudo-random generator is always set to the same value when the system is started, so that on consecutive runs the same sequences of random numbers can be expected. However, the seed can be reset as a side effect of calling RAND with special parameters.

- | | | |
|-----|---|---|
| 8.1 | GAUSS
Syntax:
Results: | Make a random number with Gaussian distribution
GAUSS(number1, number2)
A random number with gaussian distribution. The distribution is centred about the first argument, and has standard deviation by the second argument. |
| | Example: | The fictitious values in the spreadsheet AgeandIQ in the Examples directory are calculated using GAUSS(). |
| 8.2 | RAND
Syntax:
Result:

Comments: | Make a random number with rectangular distribution, or set the random number seed
RAND(number)
If the argument is positive, a random number equally likely to have any value between zero and the given argument. If the argument is negative or zero, the function returns zero.

If the given argument is zero, a side-effect is to set the random number seed to a repeatable value. If the argument is negative, the side-effect is to set the random number seed to an unrepeatable value calculated from the system clock. However, note that repeated calls with negative arguments will not give a good random sequence. |

Part 9 - Statistical functions

Introduction

The functions in this group all help to estimate missing data by fitting regression lines to existing data, using the least-squares criterion to obtain the best fit to a set of data points (x_0, y_0) , (x_1, y_1) , ... (x_n, y_n) , where the x 's are values of the independent variable (usually exact and known in advance) whilst the y 's represent the dependent observations.

There are two basic models: linear and exponential.

The linear model assumes that the best fit to the data points is a straight line of the form

$$y = mx + c$$

where m and c are constant coefficients determined by the least-squares process.

The exponential model assumes that the best fit to the group of points is an exponential curve of the form

$$y = ae^{bx}$$

where a and b are constant coefficients.

The first parameter in all the statistical functions is a block which contains the points to be fitted. The points may be arranged either in a column or in a row.

If the block is a single column (or row) it is deemed to contain y -values, and the corresponding x -values are 0,1,2,3 ...

If the block has two columns(or rows) then each row(or column) specifies a point in the data set.

For example, consider the data set (0,5), (1,8), (2,9), (3,11). This could be specified in four different ways:

- As a 4*1 block:

```
5
8
9
11
```

- As a 1*4 block:

```
5 8 9 11
```

- As a 4*2 block:

```
0 5
1 8
2 9
3 11
```

- As a 2*4 block:

```
0 1 2 3
5 8 9 11
```

A 2*2 block is deemed to have its x values along the top.

The second parameter in all the statistical functions selects the model to be fitted: 0 uses the linear model, and 1 uses the exponential model. The exponential model cannot handle zero or negative points.

Warning: Regression lines can be grossly misleading when used on unsuitable data. Before fitting a regression line examine your data points with a scatter plot or line graph.

9.1 GRADIENT and INTERCEPT

(These two functions are described together)

Syntax: GRADIENT(block, model)
INTERCEPT(block, model)

Result: The two arguments are:

- A block which gives a set of data points
- A number (0 or 1) which specifies whether the model to be fitted is linear or exponential.

These two functions serve as a pair to compute the constants of the best least-squares fit to the points supplied. The linear model uses the equation

$$y=mx+c$$

GRADIENT returns the value of **m**, and INTERCEPT, the value of **c**. The exponential model uses the equation $y = ae^{bx}$. In this case GRADIENT returns **b**, and INTERCEPT returns **a**.

Example: Suppose the sheet holds the following data:

	B	C	D	E	F
1	84	85	86	87	89
2	9.2	9.6	10.4	13.5	14.2

Then : **INTERCEPT(B1..F2,0) = -112.03**
GRADIENT(B1..F2,0) = 1.427

This suggests that the best straight-line fit for the data is

$$y = -112.03 + 1.427 * x$$

Similarly,

INTERCEPT(B1..F2,1) = 0.0001011
GRADIENT(B1..F2,1) = 0.13417

This indicates that the best exponential fit is

$$0.0001011 * e^{(0.13417 * x)}$$

9.2 PREDICTION

Estimate value of unknown datum

Syntax: PREDICTION(block, model, value)

Result: The three arguments are

- A block which gives a set of data points
- A number (0 or 1) which specifies whether the model to be fitted is linear or exponential
- A value of the independent variable.

The function fits a curve (linear or exponential) to the data set supplied as the first argument, and then uses this curve to estimate the unknown value which corresponds to the given independent variable.

Example: Suppose the sheet holds the following data:

	B	C	D	E	F
1	84	85	86	87	89
2	9.2	9.6	10.4	13.5	14.2

(Note that an entry for (x=88) is missing)

PREDICTION(B1..F2,0,88) takes a linear model and gives **13.549** as the best estimate for the y-value at x=88.

Part 10 - Financial functions

Introduction

The financial functions are all concerned with the transfer of money at regular intervals in 'contracts' which last for an extended period of time. The underlying assumptions are as follows:

- The interest rate remains constant for the whole of a contract.
- Payments are made at regular intervals - typically a month or a year. The interest rate is expressed as a fraction per period. For example, consider an annual rate of interest of 15%. This would be written as 0.15 for a contract with yearly periods.
- Payments received are recorded as positive numbers.
Payments made are recorded as negative numbers.
- Payments are made at the end of each period.
- Interest is compounded at the end of each period. For example, £1000 invested at 15% is worth:

$$1000 * (1+0.15) = 1150 \text{ at the end of the first year}$$

$$1150 * (1+0.15) = 1322.50 \text{ at the end of two years, and so on.}$$

Similarly a promise to receive £1000 with a discount of 12%, is worth

$$1000 / (1+0.12) = £892.86 \text{ a year before payment is due and}$$

$$892.86 / (1+0.12) = £797.20 \text{ two years before payment is due.}$$

Note: If the annual rate of interest is X% then the monthly rate of interest is **not** X/12%. It is actually

$$((1 + X/100)^{(1/12)} - 1) * 100 \%$$

Thus a 15% annual rate is equivalent to a 1.17% monthly rate (not 1.25%). Actually the real monthly rate for 15% annual interest using the above calculation is 1.1715 when specified to 4 decimal places. Some of the following examples use this rate.

10.1	FV	Future value
	Syntax:	FV(interest, term, payment)
	Result:	The three arguments are • The interest rate (per period) • The (integral) number of payments • The amount of each payment as a negative number.
	Comment:	The function returns the value of the investment at the end of the contract, assuming that payments are made at the start of each period.
	Example:	You invest £1000 every year for 20 years, at a promised interest rate of 27%. How much will you receive at the end of the contract? FV(0.27,20,-1000) = 555717.26

10.2	IPMT Syntax: Result: Comment: Example: Comment:	Interest component IPMT(rate, period, term, starting value) The four arguments are: • The interest rate per period • A period number (starting at 1) • The number of periods in the contract • The starting value of the contract. The function assumes that you have borrowed the starting value, and are paying equal instalments over a number of periods. This function returns the value of the interest component in any period. You have borrowed £75000 at 15% interest, repayable over 300 months. The interest component of the first payment is given by IPMT(0.011715,1,300,75000) = -878.62 The interest component of the last payment is given by IPMT(0.011715,300,300,75000) = -10.49 See the comment under PMT (10.6)
10.3	IRR Syntax: Result: Example:	Equivalent rate of interest IRR(block) The argument is a block which consists of a single row or column. Each entry records a payment, where the payments are made at regularly spaced times during the contract. There must be at least one entry of each sign. The function returns the equivalent rate of interest over the whole contract. You invest £7500 with a tunnel-building company. The return you are promised is zero at the end of each of the first five years, £2000 at the end of each of the next three years and then £3000 at then end of each of the next five years. Set up: A1 = -7500 A2..A6 = 0 A7..A9 = 2000 A10..A14 = 3000 IRR(A1..A14) gives 0.113316, or an effective rate of 11.33%.
10.4	NPER Syntax: Result: Comment: Example:	Number of payments to settle a contract NPER(interest, payment, starting value) The arguments are: • An interest rate (per period) • The amount of each payment • The initial value of the contract. The function calculates how many payments are needed to discharge the contract. You borrow £5000 at 36% interest. Assuming you plan to repay £2000 per year, how long will this take? NPER(0.36, -2000,5000) = 7.45 (just under 7.5 years)

10.5	NPV Syntax: Result: Comment: Example:	Net present value of an investment NPV(interest, block) The two arguments are: • An interest rate • A block of data which specifies payments made at regular intervals. The block must be a single row or a single column. The function calculates the net present value of the contract. Suppose you plan to invest £900 in a betting syndicate, and they promise to pay you: £100 at the end of the first year £200 at the end of the second year £500 at the end of the third year £800 at the end of the fourth year. Set A1 = 100, A2 = 200, A3 = 500, A4 = 800 If the prevailing interest rate is 18%, NPV(0.18, A1..A4) gives 945.33 which shows that the contract is marginally better than putting your money in a bank at 18% (provided you ignore the risks!).												
10.6	PMT Syntax: Result: Comment: Example: Comment:	Total payments to discharge a contract PMT(interest, term, starting value) The three arguments are: • The interest rate per period • The number of periods in the contract • The starting value of the contract. This function assumes that you have borrowed the starting value, and are paying equal instalments over a number of periods. The function returns the total amount you need to pay in any period. You have borrowed £75000 at 15% annual interest, repayable over 300 months/ Your monthly payment is given by PMT(0.011715,300,75000) = -906.151049 or £906.15. For any contract which involves regular fixed repayments, PMT will calculate the repayment required. Then for any period, IPMT will give the interest component of the payment, and PPMT the principal component. At first, since there is a large outstanding debt, the interest component will be high and the principal component low; but later this balance will change. However, at any stage the interest component and the principal component add up to the total monthly payment. This is illustrated by our example for IPMT, PPMT and PMT, where the total monthly repayments are £906.15. We find: <table> <tr> <td></td><td>First month</td><td>Last month</td></tr> <tr> <td>Interest payment</td><td>878.62</td><td>10.49</td></tr> <tr> <td>Principal payment</td><td>27.53</td><td>895.66</td></tr> <tr> <td>Total</td><td>906.15</td><td>906.15</td></tr> </table>		First month	Last month	Interest payment	878.62	10.49	Principal payment	27.53	895.66	Total	906.15	906.15
	First month	Last month												
Interest payment	878.62	10.49												
Principal payment	27.53	895.66												
Total	906.15	906.15												
10.7	PPMT Syntax: Result: Comment:	Capital component PPMT(rate, period, term, starting value) The four arguments are: • The interest rate per period • A period number (starting at 1) • The number of periods in the contract • The starting value of the contract. The function assumes that you have borrowed the starting value, and are paying equal instalments over a number of periods. This function returns the value of the capital component in any period.												

	Example:	You have borrowed £75000 at 15% interest, repayable over 300 months. The capital component of the first payment is given by $\text{PPMT}(0.011715, 1, 300, 75000) = -27.52605$ The capital component of the last payment is given by $\text{PPMT}(0.011715, 300, 300, 75000) = -895.6584$
	Comment:	See the comment under PMT (10.6)
10.8	PV Syntax: Result:	Present value of an annuity $\text{PV}(\text{interest, term, payment})$ The three arguments are • The interest rate (per period) • The (integral) number of payments • The amount of each payment (as a negative number).
	Comment:	The function calculates how much you could borrow now, given an interest rate and fixed repayments for a known number of periods.
	Example:	You wish to borrow some money to be repaid on a regular yearly basis. Given an annual rate of 27% you plan to repay £1000 a year for 20 years. How much could you borrow? $\text{PV}(0.27, 20, -1000) = 3672.62$
10.9	RATE Syntax: Result:	Interest rate for an investment $\text{RATE}(\text{term, payment, starting value})$ The three arguments are: • The number of periods of repayment • The (fixed) payment at the end of each period • The initial value of the contract.
	Comment:	The function calculates the effective interest rate of the contract.
	Example:	You have borrowed £18000, and will be repaying it at £2000 per year for 20 years. What is the interest rate? $\text{RATE}(20, -2000, 18000) = 0.092$ or 9.2%
10.10	SLN Syntax: Result:	Straight-line depreciation $\text{SLN}(\text{cost, salvage, life})$ The arguments are: • The original cost of an item • Its eventual salvage value • The life-span of the item, in periods.
	Comment:	The function calculates the 'straight line' depreciation of the item for each period of its life.
	Example:	A computer costs you £1000. It will last 4 years and then have a salvage value of £50. The annual depreciation is given by: $\text{SLN}(1000, 50, 4) = 237.50$

Part 11 - Matrix functions

Note: Matrix functions allow matrix algebra to be carried out over a spreadsheet. Matrices reside in blocks of the appropriate dimensions, one number to a cell.

Matrix functions are unlike other standard functions because they produce results which are sent to a whole group of cells at the same time. Every matrix operation has a parameter which is the identity of the block where the result is placed. This may but need not be the same as one of the operand matrices.

Each matrix function also generates a value, which is placed in the cell from which the standard function was called. The value is always zero if the operation took place correctly, or a fault indicator.

In matrix operations all operands except block operands are converted to numbers.

To illustrate, consider the function call

MATMULT(A1..D4,A5..D8, A5..D8)

which is placed in cell E7. The function multiplies the two 4x4 matrices in A1..D4 and A5..D8, and puts the result in A5..D8, obliterating the second operand. If the multiplication is successful, E7 is given the value 0; otherwise (for example, if arithmetical overflow occurs) a fault is displayed.

Normally all block arguments must be the right size, but it is permitted to specify the destination block as a one-cell block, and the system will make the right adjustments. For example, the call above could have been written:

MATMULT(A1..D4,A5..D9, A5..A5)

When a destination block is filled in, all existing data is destroyed but effects are retained.

Because of their unusual nature, matrix operations bypass the normal Intelligent evaluation mechanism. If a cell is altered as a result of matrix operation, dependent cells are not calculated automatically.

11.1	MATADD	Matrix addition
	Syntax:	MATADD(source A, source B, destination)
	Result:	Adds A and B, placing the result in the destination
	Example:	MATADD(A1..B9, G7..H15, J1..J1)
	Comment:	Matrices A and B must have the same dimensions
11.2	MATCLEAR	Clear a matrix
	Syntax:	MATCLEAR(destination)
	Result:	The destination matrix is cleared to zeros.
	Example:	MATCLEAR(A1..D9)
	Comment:	A multi-cell destination cannot be given as a single cell.
11.3	MATCOPY	Matrix copy
	Syntax:	MATCOPY(source A, destination)
	Result:	Place a copy of A in the destination.
	Example:	MATCOPY(A1..H9, J1..J1)
11.4	MATDIAG	Set a diagonal matrix
	Syntax:	MATDIAG(number,destination)
	Result:	The destination block has its leading diagonal set to the given number. and all other elements set to zero.
	Comment:	A multi-cell destination cannot be given as a single cell. The destination must be square; that is, it must have as many rows as there are columns.
11.5	MATINIT	Initialise a matrix
	Syntax:	MATINIT(number,destination)

	Result:	The destination block has all its elements set to the given number.
	Comment:	A multi-cell destination cannot be given as a single cell.
11.6	MATMULT	Matrix multiplication
	Syntax:	MATMULT(source A, source B, destination)
	Result:	Forms A*B, placing the result in the destination.
	Example:	MATMULT(A1..B9, F7..J8, P1..P1)
	Comment:	Matrices A and B must compatible dimensions: The number of columns in A must be the same as the number of rows in B.
11.7	MATSUB	Matrix subtraction
	Syntax:	MATSUB(source A, source B, destination)
	Result:	Subtract B from A, placing the result in the destination.
	Example:	MATSUB(A1..B9, G7..H15, J1..J1)
	Comment:	Matrices A and B must have the same dimensions
11.8	MATSOLVE	Solve linear equations
	Syntax:	MATSOLVE(source A, source b, destination)
	Result:	The arguments are: • Source A is a square matrix of coefficients • Source b is a column of right-hand-sides • Destination is a column of cells which is used for the result x.
	Comment:	The function solves the equations A.x = b and places the result in the destination column.
	Example:	To solve the equations $3x + 2y + z = 20$ $x + y + 2z = 19$ $2x - y + z = 11$ 1. Set the array of coefficients $\begin{matrix} 3 & 2 & 1 \\ 1 & 1 & 2 \\ 2 & -1 & 1 \end{matrix}$ into block A1..C3. 2. Set the column of right-hand sides $\begin{matrix} 20 \\ 19 \\ 11 \end{matrix}$ into block D1..D3 3. Enter the formula $=MATSOLVE(A1..C3,D1..D3,E1..E3)$ into A4 (or any blank cell except E1..E3). 4. Read the result $\begin{matrix} (x=3) \\ (y=2) \\ (z=7) \end{matrix}$ from column E1..E3.
	Comment:	This function will generate an error if the matrix of coefficients is singular.
11.9	MATTRANS	Matrix transposition
	Syntax:	MATTRANS(source, destination)
	Result:	Transposes the source matrix and places result in the destination.
	Example:	MATTRANS(A1..A10, B1..K1)
	Comments:	The destination matrix should have as many columns as the source has rows, and vice versa.

Part 12 - Miscellaneous functions

- 12.1 **ISERROR** Is a value an error?
Syntax: ISERROR(value or expression)
Result: Returns 1 if the value is an error; otherwise 0.
Examples: Put the formula `=1/1` into cell A1, and `=ISERROR(A1)` into A2. A1 will show 1, and A2, 0.
 Now change the formula in A1 to `=1/0`. Cell A1 now shows an error, **division by zero**, and A2 shows 1.
 The following formula substitutes a different message from the normal one if there is an error calculating ASIN(A1):
 IF ISERROR(ASIN(A1)) THEN "Bad angle" ELSE ASIN(A1) ENDIF
- 12.2 **TYPE** Discover type of a value
Syntax: TYPE(value or expression)
Result: The function generates a code which depends on the type of the value:
- | | |
|---------------|---|
| Integer : | 1 |
| Real number : | 2 |
| String : | 3 |
| Fault: | 4 |
| Block: | 5 |
| Image: | 6 |
- Example:** TYPE(A1)

Part 13 - Lookup Functions

- 13.1 **MATCH** Exact table look-up
- Syntax:** MATCH(number,table block,reply block)
- Result:** The arguments are:
- A value to be looked up in the table
 - A one-dimensional block: the table to be searched
 - A one-dimensional block: a list of translations.
- Comment:** The macro searches down (or along) the table, looking for an exact match to the given value. If found, it returns the corresponding entry (the one in the same position) from the list of translations; if not, it returns the string **"Value not found"**. The two blocks must each have the same number of dimensions, but they may (independently) be vertical or horizontal.
- The table is searched exhaustively, from the top or the left, and scanning stops as soon as an exact match is found. There are no restrictions on the ordering of items. The value may be a string or a number.
- Example:** Consider the data:

	A	B	C	D	E	F
1	3	5	7	12	17	18
2	Aberdeen	Dyce	Elgin	Forres	Inverness	Thurso
3	Edinburgh					
4	Newcastle					
5	Leeds					
6	London					

A call of **MATCH(7 , A1..F1 , A2..F2)** produces *Elgin*

A call of **MATCH("Thurso" , A2..F2 , A1..A6)** produces *London*

A call of **MATCH("Leeds" , A1..A6 , A2..F2)** produces *Inverness*

A call of **MATCH("Timbuctoo" , A1..A6 , A2..F2)** produces *Value not found*

13.2	SEARCH	Approximate table lookup												
	Syntax:	SEARCH(value,table block,reply block,flag)												
	Result:	The arguments are: • A value to be looked up in the table • A one-dimensional block: the table to be searched. It must be in ascending order. • A one-dimensional block: a list of translations • A flag : -1 or +1.												
	Comment:	The macro searches down the table, looking for the nearest match to the given value. If the flag is -1, the macro finds the largest entry not exceeding the given value. An exception, Value not found is returned if all the entries exceed the value. If the flag is +1, the opposite is the case; the macro looks for the smallest entry which is not less than the given value, and returns the exception if all the entries are smaller. If the macro finds an acceptable entry in the table, it returns the corresponding entry from the list of translations. The two blocks must each have the same number of dimensions, but they may (independently) be vertical or horizontal. The table is scanned from the top or the left, using the 'binary chop' algorithm. The entries in the search table must be in strictly increasing order; if they are not the result of the macro is undefined. This can be achieved by Sorting the table making sure the order is set to ascending. The value may be a string or a number.												
	Example:	Consider the data shown on the previous page: <table><tr><td>A call to SEARCH(12 , A1..F1 , A2..F2 , -1)</td><td>gives <i>Forres</i></td></tr><tr><td>A call to SEARCH(12 , A1..F1 , A2..F2 , +1)</td><td>gives <i>Forres</i></td></tr><tr><td>A call to SEARCH(13 , A1..F1 , A2..F2 , -1)</td><td>gives <i>Forres</i></td></tr><tr><td>A call to SEARCH(13 , A1..F1 , A2..F2 , +1)</td><td>gives <i>Inverness</i></td></tr><tr><td>A call to SEARCH(0 , A1..F1 , A2..F2 , -1)</td><td>gives <i>Value not found</i></td></tr><tr><td>A call to SEARCH ("Leeds" , A1..A6 , A2..F2 , 1)</td><td>would be invalid and give undefined results since the items in block A1..A6 are not in order.</td></tr></table>	A call to SEARCH(12 , A1..F1 , A2..F2 , -1)	gives <i>Forres</i>	A call to SEARCH(12 , A1..F1 , A2..F2 , +1)	gives <i>Forres</i>	A call to SEARCH(13 , A1..F1 , A2..F2 , -1)	gives <i>Forres</i>	A call to SEARCH(13 , A1..F1 , A2..F2 , +1)	gives <i>Inverness</i>	A call to SEARCH(0 , A1..F1 , A2..F2 , -1)	gives <i>Value not found</i>	A call to SEARCH ("Leeds" , A1..A6 , A2..F2 , 1)	would be invalid and give undefined results since the items in block A1..A6 are not in order.
A call to SEARCH(12 , A1..F1 , A2..F2 , -1)	gives <i>Forres</i>													
A call to SEARCH(12 , A1..F1 , A2..F2 , +1)	gives <i>Forres</i>													
A call to SEARCH(13 , A1..F1 , A2..F2 , -1)	gives <i>Forres</i>													
A call to SEARCH(13 , A1..F1 , A2..F2 , +1)	gives <i>Inverness</i>													
A call to SEARCH(0 , A1..F1 , A2..F2 , -1)	gives <i>Value not found</i>													
A call to SEARCH ("Leeds" , A1..A6 , A2..F2 , 1)	would be invalid and give undefined results since the items in block A1..A6 are not in order.													

Part 14 - Database operations

The macros described in this section are mostly similar to functions provided with Lotus 1-2-3, and permit many spreadsheets to be transferred between Lotus 1-2-3 and Schema2 without major change. The macros may also be useful to Schema2 users in their own right.

All macros modelled on Lotus 1-2-3 functions start with the characters **L_**. The rest of the name is the Lotus 1-2-3 designation.

Be warned that the database macros are not entirely compatible with Lotus 1-2-3; in particular, formulas in the criterion block are not supported, but have been replaced by relations. Furthermore the DSELECT macro is unique to Schema2 and is not available under Lotus 1-2-3.

Background

The database macros deal with two kinds of item: a **database** and a **criterion block**.

A **database** is a collection of records, all of the same shape, occupying consecutive rows of the sheet. The first row of the database holds field titles. Here is a simple database:

	A	B	C	D	E
1	First-name	Surname	Age	Sex	Place
2	George	Shaw	92	M	Office
3	Thomas	Eliot	45	M	Office
4	Mary	Shelley	18	F	Annexe
5	Elizabeth	Barrett	34	F	Office
6	Richard	Francis	51	M	Workshop
7	Joanna	Trollope	26	F	Annexe
8	Charlotte	Bronte	21	F	Workshop
9	William	Shakespeare	34	M	Office

A **criterion block** is a block of entries used to determine whether a particular record from the database is to be included in the current operation.

The first row of the block consists of some column headings from the database. The block can be as wide as you like; the column headings can be in any order, and there may be repetitions.

Each subsequent row contains one or more conditions under the appropriate headings. The condition can take one of two forms:

- It can be a string. This means that the entry in the database under this column must be an exact match to the string if the item is to be selected (subject to wild-cards, which are described below)

For example, the criterion block shown on the right means that to be selected, a record must have the string **Office** in its **Place** column.

- It can be a relation. A relation is one of the sequences LT, LE, NE, EQ, GE, or GT followed by a number. The sequences stand for

LT	Less than
LE	Less than or equal
NE	Not equal
EQ	Equal
GE	Greater or equal
GT	Greater than

The meaning is that the entry in the database under this column must be in the stated relationship to the number in the relation, if the record is to qualify.

To give an example, the block shown on the right selects all records where the entry in

Age
GE20

the **Age** column is 20 or more.

If the criterion block contains two or more conditions on the same line, they all have to be true for the record to be selected. For example, this block will select males over 50:

Sex	Age
M	GT50

The block on the right selects people whose ages are between 20 and 40 (inclusive)

Age	Age
GE20	LE40

If the criterion block holds conditions on **different** lines, each line is treated as an independent set of conditions, and allows the record to be selected if the conditions hold. For example, this block selects any record with a surname of Eliot or Shakespeare.

Surname
Eliot
Shakespeare

Age	Sex
GE65	
GE50	F

and this one selects anyone over 65 **or** any female over 50.

When the condition is a **string**, it may include wild-cards. There are two sorts:

- A **?** matches any single character.
- A ***** matches any sequence of characters. If a ***** is included in a condition, it must be the last character.

To give some examples:

"**C?p**" matches Cap, Cop, Cup
 "**????**" matches any four-character string.
 "**A***" matches any string which begins with A

In the database, the criterion

Surname
Sha*

selects the records for George Shaw and William Shakespeare.

- 14.1 **DSELECT** Determine if an entry from a database fulfils the criterion
Syntax: DSELECTION(block, criterion block)
Result: The parameters are:
- One row from a database
 - A criterion block.
- The macro returns 1 if the criterion block is true for the given row; otherwise 0.
Example: Consider the database given above and the criterion block

	J
1	AGE
2	LE21

Now suppose the formula

=DSELECT(B1..E1, @J@1..@J@2)

is recorded in block **F2..F9**. Then cells F4 and F8 will be loaded with 1's, because Mary Shelley and Charlotte Bronte are 21 or less; the other cells in the column will be loaded with zeros.

One way of using this macro is to fill a column with a formula like the one above and then sort on that column. This will bring all the selected records to the top of the database.

- 14.2 **L_DAVG** Database average
Syntax: L_DAVG(database, offset, criterion block)
Result: The parameters are

		• A database block • An offset into this block • A criterion block. The offset into the block selects a column (the columns start at 0). The macro returns the average value of the contents of the offset column in all the selected records. Example: Suppose the database is the one shown earlier, and the criterion, in cells A12..A13, is Sex M The call L_DAVG(A1..E9,2,A12..A13) gives the average age of the males: 55.5.
14.3	L_DCOUNT	Database count Syntax: L_DCOUNT(database, offset, criterion block) Comment: This macro is identical to L_DAVG (see 14.2) except that it returns the count of selected records. A column number must be given as a second parameter but is not used. Example: Assuming the same conditions as for the example in L_DAVG, a call to L_DCOUNT(A1..E9,2,A12..A13) gives 4, the number of males.
14.4	L_DMAX	Database maximum Syntax: L_DMAX(database, offset, criterion block) Comment: This macro is identical to L_DAVG (see 14.2) except that it returns the maximum value of the chosen column in the selected records. A column number must be given as a second parameter Example: Assuming the same conditions as for the example in L_DAVG, a call to L_DMAX(A1..E9,2,A12..A13) gives 92, the age of the oldest male.
14.5	L_DMIN	Database minimum Syntax: L_DMIN(database, offset, criterion block) Comment: This macro is identical to L_DAVG (see 14.2) except that it returns the minimum value of the chosen column in the selected records. A column number must be given as a second parameter. Example: Assuming the same conditions as for the example in L_DAVG, a call to L_DMIN(A1..E9,2,A12..A13) gives 34, the age of the youngest male.
14.6	L_DSTD	Database standard deviation Syntax: L_DSTD(database, offset, criterion block) Comment: This is a database function, identical to L_DAVG (see 14.2) except that it returns the standard deviation of the values in the chosen column in the selected records. A column number must be given as a second parameter. Example: Assuming the same conditions as for the example in L_DAVG, a call to L_DSTD(A1..E9,2,A12..A13) gives 21.94, the standard deviation of the ages of all males in the database.

-
- 14.7 **L_DSUM** Database sum
Syntax: L_DSUM(database, offset, criterion block)
Comment: This macro is identical to **L_DAVG** (see 14.2) except that it returns the sum of the values in the chosen column in the selected records. A column number must be given as a second parameter.
Example: Assuming the same conditions as for the example in L_DAVG, a call to
 L_DSUM(A1..E9,2,A12..A13)
 gives 222, the sum of the ages of all males in the database.
- 14.8 **L_DVAR** Database variance
Syntax: L_DVAR(database, offset, criterion block)
Comment: This is a database function, identical to **L_DAVG** (see 14.2) except that it returns the variance of the values in the chosen column in the selected records. A column number must be given as a second parameter.
Example: Assuming the same conditions as for the example in L_DAVG, a call to
 L_DVAR(A1..F8,2,A12..A13)
 gives 481.36, the variance of the ages of all males in the database.

Part 15 - Useful Macros

Note: This section deals with macros which simulate Lotus 1-2-3 functions and are intended mainly to support spreadsheets transferred from Lotus 1-2-3 but Schema2 users may find the macros useful as they stand.

Group 1: Financial macros

- 15.1 **L_DDB** Depreciation
Syntax: L_DDB(cost, salvage, life, period)
Results: The arguments are:
 - The original cost of an item
 - Its salvage value
 - Its life
 - The period of depreciation.**Comments:** Calculates depreciation through the double declining balance method. The depreciation for any given period is
(2 * book value for the period) / life of the asset
This figure is adjusted to ensure that the book value is never less than the salvage.
Example: You buy a computer for £1500. Its estimated life is 4 years, after which the scrap value will be £150. Depreciation during each of the four years is given by:
(Year 1) =L_DDB(1500,150,4,1) (750)
(Year 2) =L_DDB(1500,150,4,2) (375)
(Year 3) =L_DDB(1500,150,4,3) (187.5)
(Year 4) =L_DDB(1500,150,4,4) (37.5)
The total depreciation over 4 years is £1350, as expected.
- 15.2 **L_SYD** Accelerated depreciation
Syntax: L_SYD(cost, salvage, life, period)
Results: The arguments are:
 - The original cost of an item
 - Its salvage value
 - Its life
 - The period of depreciation.**Comments:** Calculates depreciation using an accelerated depreciation method The depreciation for any given period is
((cost - salvage)*(life - period+1)) / (life*(life+1)/2)
Example: You buy a computer for £1500. Its estimated life is 4 years, after which the scrap value will be £150. Depreciation during each of the four years is given by:
(Year 1) =L_SYD(1500,150,4,1) (540)
(Year 2) =L_SYD(1500,150,4,2) (405)
(Year 3) =L_SYD(1500,150,4,3) (270)
(Year 4) =L_SYD(1500,150,4,4) (135)
The total depreciation over 4 years is £1350, as expected.

Group 2: String macros

- 15.3 **L_FIND** Lotus Find
Syntax: L_FIND(string 1, string2, starting position)
Result: The three arguments are:
 - A needle string
 - A haystack string
 - A starting position.The macro searches string 2 (the haystack string) for an occurrence of string1 (the needle string), starting at the given position (this first character in string1 is at position 0). If found the function returns the position of the match; otherwise the string "ERR".
Example: L_FIND("ra","Abracadabra", 4) gives 9
- 15.4 **L_PROPER** Tidy a string
Syntax: L_PROPER(string)

	Result:	The macro forces the first letter of each word, in the string, into upper case and all the other letters into lower case.
	Example:	L_PROPER(" how aBOut a DRINK tHen?") gives How About A Drink Then?
15.5	L_REPLACE	Replace a segment of string
	Syntax:	L_REPLACE(original string, start, length, new string)
	Result:	The four arguments are: • original string • start position (integer, starting from zero) • a length (integer) • a new string. A segment of the original string is identified by the start position and the length. It is removed and replaced by the new string. If the length is given as zero, the new string is inserted. If the starting position is beyond the end of the original string, the new string is concatenated.
	Examples:	L_REPLACE("Computer",3,1,"") gives Comuter L_REPLACE("Computer",4,4,"troller") gives Comptroller L_REPLACE("Computer",100,0," Science") gives Computer Science
15.6	L_REPEAT	Construct repeating string
	Syntax:	L_REPEAT(string, n)
	Results:	The given string, repeated n times.
	Example:	L_REPEAT("* - ",6) gives * - * - * - * - * -
15.7	L_TRIM	Remove redundant spaces from a string.
	Syntax:	L_TRIM(string)
	Result:	The macro removes all spaces at the beginning and the end of the string, and all but the first space between words.
	Example:	L_TRIM(" A Midsummer Night's Dream ") gives A Midsummer Night's Dream

Group 3 : Miscellaneous functions

15.8	L_INDEX	Indexing a table
	Syntax:	L_INDEX(table,x,y)
	Results:	The parameters are • A block of data • x, a horizontal offset (integer) • y, a vertical offset (integer). The macro delivers the element of the table which is offset by x columns and y rows from the top-left corner.
	Example:	L_INDEX (F5..J10,2,3) gives the contents of cell H8 .
15.9	L_VAR	Variance of a block
	Syntax:	L_VAR(block)
	Result:	The macro returns the variance of the elements in the block.

Part 16 - Lotus 1-2-3 macros

Note: The macros in this section are provided solely to support spreadsheets transferred from Lotus 1-2-3. They are not of interest to users of spreadsheets native to Schema2.

Be warned that some macros with names which are similar to Schema2 functions (such as L_FV and FV) behave differently!

- | | | |
|------|--|---|
| 16.1 | L_CLEAN
Syntax:
Result: | Remove non-printing characters
L_CLEAN(string)
The string is returned unchanged. (There is no way in Schema2 to insert non-printing characters into a string). |
| 16.2 | L_CTERM
Syntax:
Result: | Time to reach future value
L_CTERM(interest, future, present)
The arguments are: <ul style="list-style-type: none">• The current interest rate per period• The present value of the investment• The future value of the investment. The macro returns the value of
$\ln(\text{future}/\text{present}) / \ln(\text{interest}+1)$ |
| 16.3 | L_DATE
Syntax:
Result: | Temporal for a given date
L_DATE(year, month, day)
The arguments are <ul style="list-style-type: none">• A year (years after 1900)• A month (1-12)• A day (1-31). |
| | Examples: | Returns the temporal for a given day in the calendar.
L_DATE(36,6,22) gives a temporal for June 22nd 1936
L_DATE(101,1,1) gives a temporal for January 1st 2001 . |
| 16.4 | L_ERR
Syntax:
Result: | Error
L_ERR()
Returns the string "ERR". |
| 16.5 | L_EXACT
Syntax:
Result: | Compare two strings
L_EXACT(string1, string2)
Returns 1 if the two string parameters are identical; otherwise 0. |
| 16.6 | L_FALSE
Syntax:
Result: | False
L_FALSE()
Returns 0. |
| 16.7 | L_FV
Syntax:
Result: | Future value of an investment
L_FV(payment, interest, term)
Calculates the future value of a regular investment made over a number of periods.
The parameters are: <ul style="list-style-type: none">• The amount of the regular payment• The interest rate per period• The number of periods. The function returns the value of the expression
$\text{payment} * ((\text{interest}+1)^{\text{periods}} - 1) / \text{interest}$ Note: This result is not the same as that returned by function FV. |
| 16.8 | L_HLOOKUP
Syntax:
Result: | Search a horizontal table
L_HLOOKUP(value, table, offset)
The parameters are : |

- A target value to be looked for
- A table, in which the top row consists of the entries to be searched. Other rows may contain related data.
- A numeric vertical offset into this table.

The macro looks for a match to the target value in the top row of the table. If the value is a string the macro insists on an exact match, but there is no constraint on the ordering of the items. If the value is a number the macro finds the largest entry in the top row which is less than or equal to the value. In this case the entries must be in increasing order.

If the macro finds a match it returns the corresponding value from the row specified by the offset parameter. If it cannot find a match it returns ERR.

Example:

Consider the following data:

	G	H	I	J	K
4	Ireland	Belgium	France	Britain	Russia
5	3500000	14500000	53000000	57000000	210000000
6	Dublin	Brussels	Paris	London	Moscow

The call **L_HLOOKUP("Belgium",G4..K6,2)** returns **Brussels**

The call **L_HLOOKUP("Britain",G4..K6,1)** returns **57000000**

The call **L_HLOOKUP("Norway",G4..K6,1)** returns **ERR**

The call **L_HLOOKUP(19000000,G5..K6,1)** returns **Brussels**

The call **L_HLOOKUP(23,G5..K6,1)** returns **ERR**

Note that the Schema2 macro lets the offset refer to areas outside the table, so that

L_HLOOKUP("Dublin",G6..K6, -2) returns **Ireland**

16.9

L_IF

Syntax:

Result:

Conditional Expression

L_IF(condition, expression 1, expression2)

The arguments are

- A condition which delivers a truth value (usually 1 or 0)
- An expression which is only used if the condition is true (non-zero)
- An expression which is only used if the condition is false (zero).

The function is used to select one of two possible expressions, depending on the conditional expression.

Examples:

L_IF(1, 2,3) gives 2

L_IF(0,2,4) gives 4

L_IF(10>9, 5*2, 11-3) gives 10

16.10

L_IRR

Syntax:

Result:

Equivalent rate of interest

L_IRR(estimate,block)

The parameters are

- An initial estimate of interest
- A block listing payments.

The function returns result identical to IRR, a standard Schema2 function. The first parameter must be present but is ignored.

16.11

L_ISERR

Syntax:

Result:

Test for errors

L_ISERR(x)

The parameter x is any expression.

The function returns 1 if x is an error marker; else 0.

16.12	L_ISNA Syntax: Result:	Tests for "Not Available" L_ISNA(x) The function returns 1 if x is the 'not available' marker; else 0.
16.13	L_ISNUMBER Syntax: Result:	Tests for a number L_ISNUMBER(x) The function returns 1 if x is a number; else 0.
16.14	L_ISSTRING Syntax: Result:	Tests for a string L_ISSTRING(x) The function returns 1 if x is a string; else 0.
16.15	L_MID Syntax: Result: Example:	Segment out a string L_MID(string, start location, size of segment) The parameters are • A string • A starting position (an integer, the first character is at zero) • The length of the segment (integer). The macro returns the stated segment. L_MID("Archimedes",2,5) = "chime"
16.16:	L_MOD Syntax: Result:	Modulus L_MOD(a,b) The two parameters are: • a number a • another number b, which must be greater than 0. The macro returns the remainder when a is divided by b.
16.17	L_N Syntax: Result:	Number L_N(block) The parameter is any block of cells. Only the top left cell is used. The function returns 0 if the top-left hand cell holds a string, or the value if the cell holds a number.
16.19	L_PI Syntax: Result:	Pi L_PI() This macro has no parameters. It returns the value of Pi correct to 15 decimal places.
16.20	L_PMT Syntax: Result:	Payments to discharge a contract L_PMT(principal, interest, term) The arguments are: • The principal, or total amount of the contract • The rate of interest per period • The number of payments to discharge the contract. The macro computes the sum payable in each period.
16.21	L_PV Syntax: Result:	Present value of an investment L_PV(payment, interest, term) This function calculates the present value of a regular income made over a number of periods. The parameters are: • The amount of the regular payment • The interest rate per period • The number of periods.
16.22	L_RAND Syntax: Result:	Random number L_RAND() This macro has no parameters. It returns a random number with rectangular distribution between 0 and 1.
16.23	L_RATE	Find interest rate

	Syntax:	L_RATE(future value, present value, periods)
	Result:	The function returns the value of the expression. (future value / present value) (1/periods) -1
16.24	L_S	String
	Syntax:	L_S(block)
	Result:	The parameter is any block of cells. Only the top left cell is used. The function returns an empty string if the top-left hand cell holds a number or the value if the cell holds a string.
16.25	L_STRING	Value as formatted string
	Syntax:	L_STRING (number, precision)
	Result:	The parameters are: • A numerical expression • A numerical precision, a whole number in the range 0 to 15. The macro returns a string which is the number converted into decimal form, with the given number of decimal digits. Rounding is applied.
	Examples:	L_STRING(1/7,5) gives "0.14286" L_STRING(L_PI(),0) gives "3"
16.26	L_TERM	Number of payment periods needed
	Syntax:	L_TERM(payment, interest, future value)
	Result:	The arguments are • A periodic payment • The rate of interest • The final value to be repaid. The macro calculates the total number of payments needed to repay a loan, using a given periodic payment. The macro returns the value of the expression $\ln(\text{future value} * \text{interest}/\text{payment})/\ln(1+\text{interest})$
16.27	L_TIME	Temporal for a given time
	Syntax:	L_TIME(hour, minute, second)
	Result:	The arguments are • An hour (0-23) • A minute (0-59) • A second (0-59). The macro returns the temporal for a given time of the day
	Example:	L_TIME(23,59,0) gives a temporal for 11.59 p.m.
16.28	L_TRUE	Truth value
	Syntax:	L_TRUE()
	Result:	Returns 1.
16.29	L_VALUE	Numerical value of string
	Syntax:	L_VALUE(string)
	Result:	The argument is any string. The macro returns the value of any number which can be extracted from the start of the string, ignoring spaces. If no number can be extracted, the macro returns 0.
	Examples:	L_VALUE(" 45 Bananas") gives 45 L_VALUE(" Schema2") gives 0.

16.30 L_VLOOKUP

Vertical table lookup

Syntax:

L_VLOOKUP(value, table, offset)

Result:

The parameters are :

- A target value to be looked for
- A table, in which the left hand column consists of the entries to be searched. Other columns may contain related data.
- A numeric horizontal offset into this table.

The macro looks for a match to the target value in the left hand column of the table. If the value is a string the macro insists on an exact match, but there is no constraint on the ordering of the items. If the value is a number the macro finds the largest entry in the left hand column which is less than or equal to the value. In this case the entries must be in increasing order.

If the macro finds a match it returns the corresponding value from the column specified by the offset parameter. If it cannot find a match it returns ERR.

Example:

Consider the following data:

	F	G	H
6	Thames	270	Britain
7	Seine	450	France
8	Volga	1200	Russia
9	Amazon	2450	Brazil
10	Nile	3200	Egypt

The call L_VLOOKUP("Volga",F6..H10,2) returns **Russia**

The call L_VLOOKUP("Nile",F6..H10,1) returns **3200**

The call L_VLOOKUP("Orinoco",F6..H10,1) returns **ERR**

The call L_VLOOKUP(3000,G6..H10,1) returns **Brazil**

The call L_VLOOKUP(23,G6..H10,1) returns **ERR**

We note that the Schema2 macro lets the offset refer to areas outside the table, so that

L_VLOOKUP("France,H6..H10, -2) will return **Seine**

Part 17 - Alphabetic list of functions and macros

Name	Type	Part
ABS	Algebraic	Part 2
ACOS	Trigonometry	Part 4
ACOT	Trigonometry	Part 4
ALL	Array	Part 6
ASC	String	Part 5
ASIN	Trigonometry	Part 4
ATAN	Trigonometry	Part 4
ATAN2	Trigonometry	Part 4
AVERAGE	Array	Part 6
CEIL	Algebraic	Part 2
CHOOSE	Database	Part 14
CHR	String	Part 5
COS	Trigonometry	Part 4
COSH	Scientific	Part 3
COT	Trigonometry	Part 4
COUNT	Array	Part 6
DAYNUMBER	Time	Part 7
DAYS	Time	Part 7
DEG	Trigonometry	Part 4
ESTDEV	Array	Part 6
EXP	Scientific	Part 3
FIND	String	Part 5
FLOOR	Algebraic	Part 2
FV	Financial	Part 10
GAUSS	Random	Part 8
GRADIENT	Statistical	Part 9
HOURS	Time	Part 7
INT	Algebraic	Part 2
INTERCEPT	Statistical	Part 9
IPMT	Financial	Part 10
IRR	Financial	Part 10
ISERROR	Miscellaneous	Part 12
L_CLEAN	Lotus macro	Part 16
L_CTERM	Lotus macro	Part 16
L_DATE	Lotus macro	Part 16
L_DAVG	Database	Part 14
L_DCOUNT	Database	Part 14
L_DDB	Useful macro	Part 15
L_DMAX	Database	Part 14
L_DMIN	Database	Part 14
L_DSTD	Database	Part 14
L_DSUM	Database	Part 14
L_DVAR	Database	Part 14
L_ERR	Lotus macro	Part 16
L_EXACT	Lotus macro	Part 16
L_FALSE	Lotus macro	Part 16
L_FIND	Useful macro	Part 15
L_FV	Lotus macro	Part 16
L_HLOOKUP	Lotus macro	Part 16
L_IF	Lotus macro	Part 16
L_INDEX	Useful macro	Part 15
L_IRR	Lotus macro	Part 16
L_ISERR	Lotus macro	Part 16
L_ISNA	Lotus macro	Part 16
L_ISNUMBER	Lotus macro	Part 16
L_ISSTRING	Lotus macro	Part 16
L_MID	Lotus macro	Part 16

Name	Type	Part
L_MOD	Lotus macro	Part 16
L_N	Lotus macro	Part 16
L_PI	Lotus macro	Part 16
L_PMT	Lotus macro	Part 16
L_PROPER	Useful macro	Part 15
L_PV	Lotus macro	Part 16
L_RAND	Lotus macro	Part 16
L_RATE	Lotus macro	Part 16
L_REPEAT	Useful macro	Part 15
L_REPLACE	Useful macro	Part 15
L_S	Lotus macro	Part 16
L_STRING	Lotus macro	Part 16
L_SYD	Useful macro	Part 15
L_TERM	Lotus macro	Part 16
L_TIME	Lotus macro	Part 16
L_TRIM	Useful macro	Part 15
L_TRUE	Lotus macro	Part 16
L_VALUE	Lotus macro	Part 16
L_VAR	Useful macro	Part 15
L_VLOOKUP	Lotus macro	Part 16
LEFT	String	Part 5
LEN	String	Part 5
LN	Scientific	Part 3
LOG	Scientific	Part 3
LOWER	String	Part 5
MATADD	Matrix	Part 11
MATCH	Lookup	Part 13
MATCLEAR	Matrix	Part 11
MATCOPY	Matrix	Part 11
MATDIAG	Matrix	Part 11
MATINIT	Matrix	Part 11
MATMULT	Matrix	Part 11
MATSOLVE	Matrix	Part 11
MATSUB	Matrix	Part 11
MATTRANS	Matrix	Part 11
MAX	Array	Part 6
MID	String	Part 5
MIN	Array	Part 6
MINUTES	Time	Part 7
MONTHADD	Time	Part 7
MONTHS	Time	Part 7
MONTHSADD	Time	Part 7
NOW	Time	Part 7
NPER	Financial	Part 10
NPV	Financial	Part 10
PMT	Financial	Part 10
PPMT	Financial	Part 10
PREDICTION	Statistical	Part 9
PRODUCT	Array	Part 6
PV	Financial	Part 10
RAD	Trigonometry	Part 4
RAND	Random	Part 8
RATE	Financial	Part 10
RIGHT	String	Part 5
ROUND	Algebraic	Part 2
SEARCH	Lookup	Part 13
SECONDS	Time	Part 7
SIGN	Algebraic	Part 2
SIN	Trigonometry	Part 4

Name	Type	Part
SINH	Scientific	Part 3
SLN	Financial	Part 10
SQRT	Scientific	Part 3
STDEV	Array	Part 6
SUM	Array	Part 6
TAN	Trigonometry	Part 4
TANH	Scientific	Part 3
TIME	Time	Part 7
TTIME	Time	Part 7
TYPE	Miscellaneous	Part 12
UPPER	String	Part 5
YEARADD	Time	Part 7
YEARS	Time	Part 7
YEARSADD	Time	Part 7

Appendix B - Definitions of TSV and CSV files

TSV Files

A TSV (Tab Separated Values) file consists of values and separators.

Each TSV value is derived from the value of a spreadsheet cell. It can be a number or a string.

When Schema2 is writing a TSV file every number is converted to characters using the number format attached to its cell. This implies that every number is written in the same format as it appears on the screen. This is useful if you just want to display the number (as in a word processor).

When Schema2 is importing a TSV file a value is interpreted as a number if it consists of the digits 0 to 9 and may include a decimal point or a leading minus sign or an optional exponent, otherwise the value is interpreted as a string.

Strings are exported and imported exactly as they appear in the spreadsheet.

The separators demarcate the values and determine which cell each value corresponds to. They can be:

- Tabs (ASCII character 9), separate cells on the same row.
- Newlines separate rows. The first value after the newline is destined for the cell at the beginning of the next row.

Example:

Annual High Jump Competition, 2023			
Name	Height /metres	Position	
Jim Dodge	1.54	2	
Sam "Big" George	1.20	3	
Graham "Lanky" Smith	1.96	1	
Average	1.57		

This spreadsheet gives the following TSV file when viewed in !Edit:

```
Annual High Jump Competition, 2023

Name[09]Height /metres[09]Position

Jim Dodge[09]1.54[09]2
Sam "Big" George[09]1.20[09]3
Graham "Lanky" Smith[09]1.96[09]1

Average[09]1.57
```

CSV Files

A CSV (Comma Separated Values) file consists of *values* and *separators*. Each value is taken from a spreadsheet cell. It can be a number or a string.

When writing a number to a CSV file Schema2 uses the full machine precision of about 15 significant figures. This is useful if you are exporting data to another application which will use it for numerical calculations.

Strings are usually enclosed in double quotes. If the string itself contains a double quote character then it is represented as two double quote characters. The backslash character introduces an Escape sequence. If the

string itself contains a backslash character it should be represented by two backslash characters. Schema2 translates all other Escape sequences into spaces

When Schema2 is importing a CSV file and finds a value which is not a number and is not enclosed in quotes it interprets it as a string.

The separators demarcate the values and determine which cell the value corresponds to. They can be:

- Commas separating cells on the same row.
- Newlines separating rows. The first value after the newline is destined for the cell at the beginning of the row.

Example:

The same spreadsheet as for the TSV example gives the following CSV file:

```
"Annual High Jump Competition, 2023"

"Name","Height /metres","Position"

"Jim Dodge",1.54,2
"Sam ""Big"" George",1.2,3
"Graham ""Lanky"" Smith",1.96,1

"Average",1.566666666666667
```

Newlines

Schema2 expects both TSV and CSV files to use the Newline character (ASCII 10) as the the line terminator. It always exports using the Newline character.

In some other environments other newline separators are used. For example in MS-DOS, it is usual for both the Newline character and the Carriage Return (ASCII 13) character to be used. Schema2 interprets any combination of Newline and Carriage Return characters when importing.

If you are exporting to a different environment you may need to modify the file in **!Edit**. To modify for MS-DOS:

1. Export from Schema2 as TSV or CSV.
2. Load into !Edit.
3. Bring up the **Find** dialogue. Enter: Find: \n Replace with: \x0D\n
4. Click select on **Go**.
5. Click select on **End of file replace**.

Appendix C - Schema2 and Lotus 1-2-3

This appendix discusses the transfer of spreadsheets between Schema2 and Lotus 1-2-3, or any other spreadsheet which can read or write data in **WK1** format.

Introduction

Schema2 and **Lotus 1-2-3** are built to a common standard. Both spreadsheets use columns labelled A B C and rows numbered 1 2 3. They both handle expressions which include constants and cell names, and have libraries of standard functions. In principle, therefore, it is not difficult to transfer a working spreadsheet from **Lotus 1-2-3** to **Schema2**, and vice versa.

The **Schema2** transfer programs assume a version of **Lotus1-2-3** which is the lowest common denominator of several variants. This implies that certain aspects of spreadsheets are not handled:

- Multi-sheet systems cannot be moved between **Lotus 1-2-3** and **Schema2** (except as collections of unrelated sheets).
- No stylistic information on fonts, character size or colours is transferred.
- No macros can be moved from one system to the other.

Recommendations for Lotus1-2-3 users:

If you intend to transfer your **Lotus 1-2-3** sheet to **Schema2**, you should follow these rules:

- The application must be on a single sheet.
- The application should use no macros.
- The application must not use the following @ functions:

@TODAY
@TIMEVALUE
@CHOOSE
@DATEVALUE
@CELL
@CALL
@APP
@@

- The application must restrict its use of functions such as @SUM, which normally take an arbitrary number of arguments. The only argument allowed to such a function is a single range. For example

@SUM(G1..J70)

will be translated correctly, but

@SUM(G1, H3, I4..J10, K11)

will not be handled in the right way.

- Note that Schema2 handles the data base functions @DSUM, @DAVG, etc., in a slightly different way from **Lotus1-2-3**. The Schema2 method is described in Appendix B, part 14.

To move a spreadsheet to **Schema2**, use your **Lotus1-2-3** (or compatible) program to file the spreadsheet in **WK1** format. Then import this file as described in Techniques 14.

Recommendations for Schema2 users

If you intend to move your **Schema2** spreadsheet to **Lotus 1-2-3** you must observe several important restrictions:

- Use only one sheet for your application.
- Use **Lotus time mode**, right from the start.

- Do not use any of your own macros.
- Use only the following standard functions and macros:

ABS	ACOS	ASC	ASIN	ATAN
ATAN2	AVERAGE	CHR	COS	COUNT
DAYS	EXP	FIRST	FLOOR	HOURS
INT				
L_CLEAN	L_CTERM	L_DATE	L_DAVG	L_DCOUNT
L_DDB	L_DMAX	L_DMIN	L_DSTD	L_DSUM
L_DVAR	L_ERR	L_EXACT	L_FALSE	L_FIND
L_FV	L_HLOOKUP		L_IF	L_INDEX
L_IRR	L_ISERR	L_ISNA	L_ISNUMBER	
L_ISSTRING	L_MID	L_MOD	L_N	L_NA
L_PI	L_PMT	L_PROPER	L_PV	L_RAND
L_RATE	L_REPEAT	L_REPLACE	L_S	L_STRING
L_SYD	L_TERM	L_TIME	L_TRIM	L_TRUE
L_VALUE	L_VAR	L_VLOOKUP		
LEFT	LEN	LN	LOG	LOWER
MAX	MIN	MINUTES	MONTHS	NOW
NPV	RIGHT	ROUND	SECOND	SECONDS
SIN	SLN	SQRT	STDEV	SUM
TAN	UPPER	YEARS		

- Techniques 15 describes how to save a **Schema2** sheet in **WK1** format on an MS-DOS disk. In this format it can be loaded directly into **Lotus 1-2-3**.

Appendix D - Notes for users transferring from Schema

This section lists some of the major differences between Schema and Schema2. We do not list all the new features of Schema2.

Filetypes

Schema2 files are saved in a different format from Schema files. See Section 14 (page 182) for instructions on importing Schema files and workspaces

Selection

Schema2 has only one selected area, which may be a single cell or a rectangular block of cells. See Section 1 (pages 111-112) for instructions on changing the selection.

Entering data and formulae

Schema2 does not have the separate edit window that Schema had. See Example 1 and Section 2 for details of entering data and formulae.

Copy and Move

Copying and Moving data in Schema2 is done by using the clipboard. See Section 10 for details of cut, copy, paste.

Styles

Styles do not have separate rules for positive, negative, zero and strings. Instead the style defines the attributes for a string with the number format defining anything that is different for numerical values. There is no current style in Schema2. The appearance of cells can be modified by applying Effects to them as well as styles. See Examples 5 and 6, and Section 10 for further details.

Graphs

The system for producing graphs is completely different in Schema2. See Sections 22-24 for details.

User names

In Schema2 User names are called Sheet variables.

Appendix E - Keyboard short cuts in Schema2

Many keyboard short cuts in Schema2 can be changed by modifying menu description files (See Part 2 Techniques 27).

In the table, the symbol $\uparrow$ refers to the **Shift** key and CTRL means the control key.

For example, to select all cells, hold down the **Control** key and press **A**.

Key	Meaning	Can be reset
CTRL+A	Select all	Yes
CTRL+B	Bold	Yes
CTRL+C	Copy	Yes
CTRL+E	Recalculate all	Yes
CTRL+F	Big cell	Yes
CTRL+I	Italic	Yes
CTRL+K	Delete	Yes
CTRL+O	Labels	Yes
CTRL+R	Icon ribbon	Yes
CTRL+T	Simple graph	Yes
CTRL+U	Clear entry area	No
CTRL+V	Paste	Yes
CTRL+X	Cut	Yes
CTRL+Y	Move	Yes
CTRL+Z	Clear	Yes
F2	Switch to mouse mode	No
F3	Save	Yes
F5	Go to cell	Yes
F6	Sort	Yes
F7	Graph	Yes
F8	Number format	Yes
F10	Grid	Yes
F11	Scale view	Yes

Key	Meaning	Can be reset
CTRL+F1	Info	Yes
CTRL+F2	Discard sheet	Yes
CTRL+F3	Save workspace	No
CTRL+F5	Left alignment	Yes
CTRL+F6	Centre alignment	Yes
CTRL+F7	Right alignment	Yes
CTRL+F8	Justify	Yes
CTRL+F10	Do manual eval	Yes
CTRL+F11	Formulae	Yes
↑F3	Scale 30%	
↑F4	Scale 40%	
↑F5	Scale 50%	
↑F6	Scale 60%	
↑F7	Scale 70%	
↑F8	Scale 80%	
↑F9	Scale 90%	
↑F10	Scale 100%	
↑F11	Scale view at half current scale	
↑F8	Input block	Yes
↑F11	Scale view at double current scale	
Print	Display Print dialogue	Yes
Cursor down	Select 1 cell down	No
Cursor left	Select 1 cell left	No
Cursor right	Select 1 cell right	No
Cursor up	Select 1 cell up	No
Home	Select first cell	No
Page down	Scroll down one window	No
Page up	Scroll up one window	No
CTRL+Cursor down	Select bottom cell	No
CTRL+Cursor left	Select left-most cell	No
CTRL+Cursor right	Select right-most cell	No
CTRL+Cursor up	Select top cell	No
↑Cursor down	Scroll down one window	No
↑Cursor up	Scroll up one window	No
↑Cursor left	Scroll left one window	No
↑Cursor right	Scroll right one window	No

Index

!	95	C	
%	83	Calls to standard functions	117
-	151	Cancelling a formula with the Cross icon	70
1. Introduction	161	CAT	117
3D	129	CEIL	162
A		Centred	112
A5 Landscape	129	Change name on save	95, 99
A5 Portrait	129	Changing a selected block	68
ABS	162	Changing keyboard short cuts	159
ACOS	164	Changing the default document	153
ACOT	164	Changing the margin content	75
Adjusting the width of a column	72	Changing the width of a column	71
Advance	90	Characters per line	111
After number	81	Choices dialogue	76, 154
Algebraic functions	125, 162	Choosing measurement units with the Choices Dialogue	71
Alignment	82, 86-87	CHR	166
ALL	168	Circle	144
Altering the icon ribbon	160	Circular references	148
Alternative for zero	83	Clear	139
An example of multiple spreadsheet working	95	Clear all effects	89
And	118	Closing a view	76
Angles	151, 153	Collated	111
Appearance of numbers	114	Colour of Negative numbers	84
Appearance of views	76	Colours	142
Applying a style	88	Column and Row Dimensions	71
Arguments and results	122	Column and Row labels	112
Arithmetic operators	117	Common Print options	111
Array functions	124, 168	Condensed	111
ASC	166	Conditional operators	119
ASIN	164	Constants	114
Aspect ratio	140, 144	Controlling spreadsheet layout	113
ATAN	164	Copies	111
ATAN2	164	Copy	90-91
Auto load	154	Copy and Move	202
Automatic and Manual Evaluation	149	COS	164
Automatic evaluation	146, 149	COSH	163
AVERAGE	168	COT	164
Axes	128	COUNT	168
Axis ticks	143	Create New Spreadsheet	153
Axis type	128	Creating a new style	88
B		Creating and destroying Explicit big cells	78
Background	184	Creating Sheet Variables	120
Background colour	87	Creation of Automatic big cells	78
Bar Width	142	Criterion block	184
Before number	81	CSV1	07-108
Big cell	104	CSV -	103
Bitwise logical operators	118	CSV Files	198
Bitwise shift operators	119	Ctrl-K	139
Black grid	154	Ctrl-Z	139
Black grid -	112	Currency	81
Block designations	116	Custom	129
Boldness	87	Custom format	80
Box	144	Custom formats	83
Brackets	117, 120	Cut	90
Bring to front	139	Cut, Copy and Paste	90

D

Data Direction	127
Database operations	184
DataPower	90
Date	81
Date and Time (Temporal) values	115
DAYNUMBER	170
DAYS	170
Decimal Places	81
Decimal Separator	82
Decimals	114
Decorating Graphs	139
Default Document	71, 73, 153-154
DEG	164
Degrees	151
Delete	90, 92, 139
Deleting a style	89
Deleting Columns	74
Deleting Items	158
Deleting Rows	73
Deleting Sheet Variables	121
Dimensions	143
Discard sheet on close	76, 154
Discarding a graph	146
Discarding Spreadsheets	76
Displaying a percent sign	83
Division by zero	117
Do manual eval	150
Draft	110
Draft printing	109
Dragging objects	139
Drawfile	107
Drawfiles	103
DSELECT	185

E

Edit submenu	139
Editing a Menu Description file	158
Editing a style	89
Editing Sheet Variables	121
Editing tools	90
Ellipse	144
Empty cell rule	153
Empty cell value	151, 168
End cap	141
Entering a function with the mouse	69
Entering a macro into a formula	69
Entering a sheet variable into a formula	70
Entering cell and block names	70
Entering data and formulae	202
Entering Dates	115
Entering Times	116
ESTDEV	168
Evaluation	148
Evaluation by row and by column	149
Evaluation mode	153
Evaluation mode dialogue	150
Exclusive Or	118
EXP	163
Explode	142
Export	107

Exporting CSV files	107
Exporting Drawfiles	107
Exporting Formulae	108
Exporting Print text	108
Exporting TSV files	107
Exporting WK1 files	107
Extra characters for negative values	85

F

File Information	100
File menu	146
File submenu	139
File types for import	103
Filetypes	202
Fill colour	140
Final comment	151
Financial functions	125, 175
FIND	166
Fit to cell	147
Fit to page	112
Fit to selection	129, 147
Fixed decimal	81
FLOOR	162
Font name	87
Font size	87
Forced newlines	79
Foreground colour	87
Formulae	77
Fraction separator	85
Frame	144
Full size	144
Function arguments	161
Function icon	69
FV	175

G

GAUSS	172
General	81
General graphs	138
Go to data	144
GRADIENT	174
Graph Axes	126
Graph Axes dialogue	128
Graph colours	142, 154
Graph Data	126
Graph Data block	127
Graph Data Dialogue	127
Graph dialogues	126
Graph Ed	147
Graph Edit menu	139
Graph Options	126
Graph Options dialogue	128
Graph Parameters	127
Graph size	129
Graph Type	127
Graph types	130
Graphs	202
Gregorian Calendar	170
Grid	77, 141, 153
Grid size	144
Group 1:Financial macros	188
Group 2:String macros	188

Group 3 : Miscellaneous functions	189	LEFT	166
		LEN	166
H		Limits	128
Hiding and restoring System labels	75	Line	144
Hiding rows and columns	113	Line colour	140-141
Hints on saving	100	Line dashes	140
Histograms	134	Line plot	130
Horizontal histograms	134	Line width	140
Horizontally	112	Linear equations	180
Hotlink	146-147	LN	163
HOURS	170	Loading	99-100
House style	87	Loading a Workspace	101
		Loading resource files with a workspace	155
I		Lock Aspect	147
Icon ribbon	77, 153, 155	Locking rows or columns	75
IF ... THEN ... ELSE ... ENDIF	119	LOG	163
Image (normal) printing	109	Logical shift left	119
Image printing	109	Logical shift right	119
Importing CSV files	104	Long strings	78
Importing data in MS-DOS format	106	Lotus 1-2-3	103, 105, 151, 154
Importing Schema Files	105	Lotus 1-2-3 macros	190
Importing Schema Workspaces	106	Lotus mode	115
Importing SID files	105	Lotus time mode	170
Importing spreadsheet data	103	LOWER	166
Importing Sprites and Drawfiles	104	L_CLEAN	190
Importing TSV files	104	L_CTERM	190
Importing WK1 files	105	L_DATE	190
Impression Style	90	L_DAVG	185
Info	100	L_DCOUNT	186
Inner border	143	L_DDB	188
Insert submenu	144	L_DMAX	186
Inserting columns	74	L_DMIN	186
Inserting rows	73	L_DSTD	186
INT	162	L_DSUM	187
Integer	81	L_DVAR	187
Integers	114	L_ERR	190
Intelligent and Mechanical evaluation	149	L_EXACT	190
Intelligent evaluation	149, 179	L_FALSE	190
INTERCEPT	174	L_FIND	188
Interest rat	175	L_FV	190
Introduction	126, 173, 175, 200	L_HLOOKUP	190
Invert bits	118	L_IF	191
IPMT	176	L_INDEX	189
IRR	176	L_IRR	191
ISERROR	181	L_ISERR	191
It menu	147	L_ISNA	192
Italicisation	87	L_ISNUMBER	192
Iteration count	150	L_ISSTRING	192
		L_MID	192
K		L_MOD	192
Keyboard short cuts	155	L_N	192
		L_PI	192
L		L_PMT	192
Labels	77, 153	L_PROPER	188
Labels with Parameters	157	L_PV	192
Landscape	111	L_RAND	192
Layout of dates and times	83	L_RATE	192
Layout of numbers	83	L_REPEAT	189
Least-squares criterion	173	L_REPLACE	189
		L_S	193
		L_STRING	193
		L_SYD	188

L_TERM	193
L_TIME	193
L_TRIM	189
L_TRUE	193
L_VALUE	193
L_VAR	189
L_VLOOKUP	194

M

Macro calls	117
Making a new view	76
Manual evaluation	149-150
Margins	112
Marked lines	131
MATADD	179
MATCH	97, 182
MATCLEAR	179
MATCOPY	179
MATDIAG	179
MATINIT	179
MATMULT	180
Matrix functions	126, 179
Matrix multiplication	180
Matrix printers	112
MATSOLVE	180
MATSUB	180
MATTRANS	180
MAX	168
Menu Description Files	155
Menus	155
Menus sub directory	155
MID	166
MIN	168
MINUTES	170
Miscellaneous functions	181
Modifying the resource files in !Schema2	160
MONTHS	170
MONTHSADD	170
More details about workspaces	101
Move	90, 92
Moving references - an exact description	92
Multi-line text	78-79
Multiple spreadshee	150
Multiple spreadsheets	95

N

Negative display	82
New Graph	126
New Graph dialogue	127
NOW	171
NPER	176
NPV	177
Number format	80, 87, 114
Number format dialogue	80
Numbers	114

O

Omitting the grid	112
Operands	117
Operators	117
Options for draft mode	111
Options for the image mode	111
Or	118
Othe	129
Outer border	143
Overall size	143

P

Page breaks	113
Paste	90-91
Paste values	90
Paste values -> Normal	91
Paste values -> Transpose	91
Paste values Transposed	90
Pictogram	160
Pictograms	129, 137, 155
Picture	129
Pictures	155
Pictures directory	155
Pie Charts	136
Plain style	75, 81
Plotting a simple graph	126
PMT	177
Pocketbook	107
Point size	143
Point style	143
Portrait	111
Power operator	117
PPMT	177
PREDICTION	174
Print	146
Print Options	110, 153
Print text	110
Printing	109
Printing a spreadsheet	110
PRODUCT	169
Pseudo-random functions	172
PV	178

R

RAD	164
Radians	151
RAND	172
Random functions	172
Random number generators	125
RATE	178
Re-opening a hotlinked graph	147
Recalculate all	98, 150
Recommendations for Lotus1-2-3 users:	200
Recommendations for Schema2 users	200
Reference to undefined cell	168
Relational operators	118
Remainder operator	117
Removing Effects	89
Reopen graph	147
Replicate	90, 92
Resource files	102, 155

Reveal	113	Squeezed	144
RIGHT	167	Stacked Horizontal histograms	135
ROUND	162	Stacked Vertical histograms	134
S		Standard Functions	122
Save	146	Standard macros	126
Save as default	153	Start cap	141
Saving	99	Statistical functions	125, 173
Saving a sheet previously loaded		STDEV	169
from a file and changed.	99	String functions	123, 166
Saving a workspace	101	String operator	117
Saving Selection	100	Strings	116
Scale	111, 153	Style	153
Scale view	77, 144	Styles	87, 202
Scaled	144	SUM	169
Scatter diagrams	131	SUM()	69
Schema -	103	T	
Schema mod	115	TAN	165
Schema time mode	170	TANH	163
Schema2\$DateFormat	81, 115	Techniques 7 - Creating and	
Schema2\$TimeFormat	81	Destroying Big Cells	78
Scientific	81	Temporal value	81, 123, 170
Scientific format	81	Temporal values	115
Scientific Functions	124, 163	Terminating a formula with	
Scientific notation	114	the Tick icon	70
Scientific numbers	114	Text	103
SEARCH	183	Text aspect	140
SECONDS	171	Text background	140
Segment colour	142	Text colour	140
Select Graph	139	Text font	140
Selecting a block	67	Text size -	140
Selecting a complete column	67	The Appearance of Row	
Selecting a complete row	67	and Column Labels	75
Selecting a set of columns	68	The Clipboard	90
Selecting a set of rows	68	The Effects Submenu	140
Selecting a single cell	67	The Function Groups	123
Selecting all the cells in the sheet	68	The Graph Submenu	141
Selecting areas to print	112	The Icon Ribbon	159
Selecting Objects	139	The meaning of a Custom format	83
Selection	202	The Modification Marker	100
Sequence	86	The need for Recalculation	98
Setting a Custom format	83	The NOT operator	118
Setting a Labels style	89	The Pictures directory	160
Setting a new width	71	The Print dialogues	110
Setting a Number format	80	Thousands separator	82, 85
Sharing styles between sheets	89	Three D effect -	141
Sheet naming convention	95	Time	81, 171
Sheet variable	150, 153	Time functions	123, 170
Sheet variables	120	Time mode	115, 123, 151, 153, 170
Shift operator	119	Time value	151
Show Selection page breaks	111	To add a new icon	160
SID -	103	Transferring files to the MS-DOS	
Sideways	111	environment.	108
SIGN	162	Transpose	91
Simple Graph	138	Triangle	141
SIN	165	Trigonometric functions	124
SINH	163	Trigonometrical Functions	164
Sizing graphs	147	True	118
SLN	178	Ts	150
Snap to grid	144	TSV	107-108
Specifying a graph	126	TSV -	103
SQRT	163	TSV Files	198

Schema2

TTIME	171
Type	81, 181
U	
Units	154
UPPER	167
Upright	111
Useful Macros	188
User names	202
Using multiple operators and precedence	119
Using multiple sheets	95
Using Sheet Variables	121
Using the clipboard to import data from other applications	106
Using the Create New Spreadsheet dialogue.	73
Using the Sum icon	69
Utilities menu	144
V	
Values	151
Vertical alignment	87
Vertical histograms	132
Vertically	112
Views	76
W	
Whole numbers	114
WK1	105, 107
WK1 -	103
Workspace	98, 155
Workspace\$Dir	102
Workspaces	99, 101
X	
X data set	127
Y	
Y axis	128
Year	84
YEARS	171
YEARSADD	171

Here are just a few of the other products you will be able to
download from RISCOS.com free of charge
(You can make a donation to help support future product releases but this is entirely voluntary)



This Documentation is © 3QD Developments Ltd 2026
Some of the products shown above might not be available immediately.
Not all products will run on all versions of RISC OS, check the product information page for details.

Other Products list last updated 29/2/26