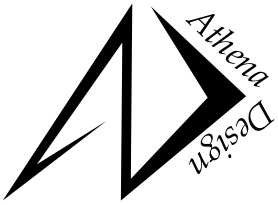
MesaBook.book : FrontMatter.frame Black #1

Mesa

User's Guide

Release 1.5



Athena Design, Inc.
17 Saint Mary's Court
Boston, MA 02146

Copyright

Neither this documentation (book) nor the software may be copied, photocopied, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, in violation of the license agreement of the software that accompanies this book.

Copyright © 1992 & 1993 Athena Design, Inc.
17 Saint Mary's Court
Boston, MA 02146

All rights reserved. First edition printed 1993.

Written by: Elizabeth Ditchburn, Greg Hudson, Linda Julien, Victor Maslov, and David Pollak
Edited by: Elizabeth Ditchburn, Greg Hudson, Tracy Kugelman, Linda Julien, David Pollak, Sue Rankin, and Ellen Ullman

Product names mentioned in this book may be trademarks and are used for identification purposes.

Introduction 1
Welcome 1
How to Use this Book 2
About Mesa 3
Technical Support 5

Special Features 7
Introduction 7
ToolBar 7
Tools on the W orksheet 9
SmartFill 10
Drag & Drop Ranges 11
Formula Inheritance 12
Graphing 13
Conclusion 15

Getting Around Mesa 16
Basic topics 17
Print topics 38
Advanced topics 42

Tutorial 60
Introduction 60
Creating the W orksheet 60
Entering the Column Headings 62
Entering the Formulas 64
Formatting 65
AutoGrow Ranges 67
Reports 68
Summary 70

Inspectors and Panels 71
Preferences Panel 72
Cell Style Inspector 75
Sheet Inspector 83
Graph Inspector 88
Shape Panel 101

Layer Inspector 101
Report Layout Inspector 102
Label Inspector 114
Query Inspector 115
Scenario Inspector 124
Extract Inspector 128
Event Inspector 129
MScript Inspector 130
Pattern Panel 131

Mesa Menus 132
The Main Menu 133
The Info Menu 137
The File Menu 139
The Edit Menu 142
The Format Menu 150
The Sheet Menu 159
The Graphics Menu 167
The W indows Menu 182
AddIn Menu 185

Functions 186
Functions and @Functions 187
Entering a Formula into a Cell 187
Function Arguments 188
Operators 192
Math Functions 198
Financial Functions 225
String Functions 240
Special Functions 251
Date Functions 277
Database Functions 285
Constants 301

MScript 306
The Language 306
Creating and Editing a Script 306

Errors 308
Basic Operations 308
Functions 318
File Operations 318
Editing Operations 320
Formatting Operations 334
Graphics Operations 364
Other Operations 369
Retrieving information about the sheet 380
Constants 391

Frequently Asked Questions ..... a

Keyboard Reference ..... g

MesaBook.book : MesaBookTOC.frame Black #4

1. Introduction

Welcome

Welcome to the *Mesa User's Guide*. This manual is intended to introduce you to spreadsheets and to Mesa, show you have to create a worksheet, and act as a reference for Mesa's menus, inspectors, functions, and scripting language. This manual covers both Mesa and Mesa Sunrise. Mesa Sunrise has all the traditional spreadsheet features of Mesa, but does not include SQL database access, and the special programming features that allow Mesa to be linked to custom applications or to add custom programs into Mesa. The parts of this manual that do not apply to Mesa Sunrise will have this symbol in the left margin:

This manual was printed with generous "white space" for better readability and to allow you to make comments and write in the margins. If you have good ideas and want to make notes in the manual, please do so. This will allow you to remember what you were thinking about as you read a certain section.

We hope that this manual teaches you about Mesa and acts as a reference guide for your future use of Mesa.

How to Use this Book

The *User's Guide* provides you with the information necessary to start Mesa and to use it on your NEXTSTEP computer. The Guide does not attempt to provide you with comprehensive information about NEXTSTEP computers or the NEXTSTEP operating environment. Please refer to the appropriate product manuals for information about these topics.

Typographical Conventions

The following conventions are used throughout this book:

Mesa menu options will appear in bold type, for example: **Format**.

Menu options located on submenus will appear as: **Graphics → Bar Chart.**

Command key equivalents, when available, will appear in parentheses after menu options, for example: **File → Save** (*Command-s*). *Command- s* means: hold down the *Command* key while typing *s*

Special keys will be in italics, for example: *Control, Alt, Shift, Command.*.

Command-Shift-S means: hold down the *Command* and *Shift* keys while typing *S*.

About Mesa

Mesa is a traditional spreadsheet program that runs on NEXTSTEP computers. Mesa looks and works similarly to Lotus 1-2-3 and Microsoft Excel. If you have used either of these programs, then Mesa should be very easy to learn. Please note that while Mesa works very much like 1-2-3 and Excel, Mesa does do things in slightly different ways or menu options will be located in different places. For example, Mesa uses a NEXTSTEP convention called *Inspectors* to achieve many results. Just because Mesa does not work the exact same way as 1-2-3 or Excel does not mean that Mesa cannot produce the same result. If have questions, please call or look in the index.

A spreadsheet is a program that allows the user to enter data into matrix of cells. Each cell can be individually addressed based on its row and column number. In Mesa, rows are numbered from 1 to 16384 and columns are labeled from A to XFD. Each cell has a value and formatting information. The cell's value is based on either its contents (e.g., a number like 5, or a string like *Hello*) or on a formula that is evaluated to result in the cell's value. A value can be a number (e.g., 1, 4.7, 5280), a string (e.g., *Hello* , *June Stock Prices* , *Eastern Region*), or an error (e.g., !ERR, !/0). A formula can refer to the value in a cell during calculation. This is why spreadsheets are so powerful. Because the value in one cell can depend on the value in another cell, when the first cell is changed, the second cell updates itself properly. For example, if a cell contained the formula *=A1+1*, when the value in cell A1 is changed, the cell containing the formula automatically updates itself.

Cells also contain formatting information. This includes the text format of the display (e.g., General number, Scientific notation, Currency format), the color of the text, the background color of the cell, and more. Information can be presented in a more visually pleasing or informative manner by changing the cell formatting. Mesa allows some of the cell's formatting to be changed by the cell's formula so that a cell can automatically change its color if it contains certain values. This feature allows worksheets to alert the user to important cells that contain critical values.

Once the cells have been populated with values, and the spreadsheet has been formatted, it can be printed out, e-mailed to others, or saved. In addition, Mesa allows you to graph the values in cells to produce presentation-quality graphs and include those graphs in reports or copy the graphs or groups of cells and paste them into other NEXTSTEP applications including Create and DataPhile from Stone Design, Concurrence from Lighthouse Design, or any NEXTSTEP application that utilizes live links.

Mesa also allows data to be exchanged with SQL database systems. Mesa can retrieve data from an SQL system. Once the data is in Mesa, it can be formatted, you can enter formulas that calculate based on the data retrieved, and you can produce presentation quality reports. When you use Mesa this way, it becomes a report writing tool for SQL database systems.

Mesa supports real time data feeds. You can purchase modules that add in to Mesa (AddIn's) that provide real time stock market data feeds, and other live information. This allows you to build a model (such as a stock portfolio), and then have

the current real world information flow into the worksheet thus keeping the portfolio values up to date.

Mesa also includes MOLI – the Mesa Object Library Interface – which allows another application to link right into the data in a worksheet, execute database queries, graph information, and much more. Many NEXTSTEP sites link their mission critical custom applications into Mesa so that Mesa can graph data for them, create a unified look to all their applications, reduce the amount of programming time to build the custom applications, and to access many of Mesa's features.

Once the data has been entered into Mesa from various different sources, the formulas have been entered so that the raw data can be summarized, the formatting has been applied, and the charts and graphs have been created, you can print the worksheet in order to produce a presentation quality report. Mesa takes advantage of Display PostScript which is built into every NEXTSTEP system to produce crisp, presentation quality output on whatever printing device you have connected to your NEXTSTEP system.

Technical Support

Commercial versions of Mesa include free unlimited technical support. Athena Design's technical support lines are open from 8:30 am to 5:30 pm Eastern time, Monday through Friday, except holidays. The technical support phone number is 1.617.734.6372. Please have your license number handy when calling for technical support. You

can also request technical support via e-mail. The e-mail address for technical support is feedback@athena.com, or you can use the **Info → Feedback** command to compose and send e-mail directly from Mesa. Mesa automatically includes the version of Mesa and your license number with mail sent from **Info → Feedback**.

Mesa Sunrise comes with 4&4 support. We will provide technical support via phone or e-mail for the first four months of Sunrise ownership. After that, we will answer four more technical support questions, no matter when you ask them. This allows you unlimited access to technical support during the first few months of using Sunrise, but at the same time if you run across a support question after using Sunrise for a year, you can still get that question answered. Athena Design's technical support lines are open from 8:30 am to 5:30 pm Eastern time, Monday through Friday, except holidays. The technical support phone number is 1.617.734.6372. Please have your license number handy when calling for technical support. You can also request technical support via e-mail. The e-mail address for technical support is feedback@athena.com, or you can use the **Info → Feedback** command to compose and send e-mail directly from Mesa. Mesa automatically includes the version of Mesa and your license number with mail sent from **Info → Feedback**.

2. Special Features

Introduction

Mesa contains many special features that make using Mesa seamless and intuitive. This section will guide you through some of these features including the Mesa ToolBar, SmartFill, Drag & Drop color, and more. If you would like to learn about basic spreadsheet functions, please skip ahead to the next chapter.

In order to perform the special features listed in this chapter, you must have installed Mesa and started the Mesa application.

ToolBar

Mesa has a customizable ToolBar onto which you can place any command, script, macro, report, SQL queries, and more. Mesa's ToolBar is special because you can populate it easily using *Direct Manipulation* .

Note: NEXTSTEP software is generally easier to use that Macintosh or W indows software because many things in NEXTSTEP are done by *Direct Manipulation* : "grabbing" the thing you want to change and placing it where you want it. This is generally eas-

ier than going into a dialog box and entering values or typing something. The NEXTSTEP color picker allows you to change colors by grabbing a piece of color and placing it where you want.

To open Mesa's ToolBar, perform the **Format** → **ToolBar...** command. To do this, click on the **Format** selection on the main menu. The **Format** submenu will appear. Click on the **ToolBar...** selection. Mesa will display the ToolBar, a three by seven array of tools. Some tools will be populated while others will be blank. You can move and resize the ToolBar to suite your work-habits.

To place a new tool on the ToolBar, Command-Drag the desired tool onto the ToolBar. As an example, let's move the **File** → **New** command onto the ToolBar. Select the **File** menu by clicking on **File** on the main menu. The **File** submenu will appear. Press down and hold the *Command* key. Click, but do not release, on the **New** selection on the **File** submenu. Notice that an icon appears under the mouse. While holding the *Command* key, drag the icon on top of one of the squares on the ToolBar. Note that as you drag the icon over a square, the square highlights. Release the mouse button when the icon is over the desired square on the ToolBar. The ToolBar will update with the new command. To execute the command, click on the button. A new worksheet will appear just as if you had selected the **File** → **New** command.

You can populate the ToolBar with any command from any menu by *Command-Dragging* the item onto the ToolBar.

Note: To clear an item off the ToolBar, *Command-Control-Drag* any item onto the item that you want to remove. *Command-Control-Dragging* creates a blank item.

Tools on the Worksheet

Just as you can *Command-Drag* an item onto the ToolBar, you can place an item onto the worksheet. This is useful if you have a command or a script that you use at a certain place in the worksheet. For example, if you have a script that validates data in a data entry in a certain range on the worksheet, you can place an MScript button near that range so that you can run the script by clicking on the button.

To place a command or a script on the worksheet, *Command-Drag* the item and drop it onto the worksheet just like you would drop the item onto the ToolBar. Now the tool works just like it does on the menu or on a ToolBar.

Click on the *New* button that you placed on the ToolBar. Mesa will create a new, blank worksheet. Select the **File** menu. *Command-Drag* the **Save** item and drop it on the worksheet. Click on the button and Mesa will prompt you for the file name to use to save this worksheet. Enter

MySheet and press ↵. Mesa will save the worksheet in your home directory with the name *MySheet.Mesa* .

Note: To remove a tool from the worksheet, *Command-Control-Drag* a blank tool and drop it onto the tool on the worksheet. To move a tool on the worksheet, *Command-Drag* the tool to a new location.

SmartFill

Mesa provides a quick way to fill ranges with sequences of information. Many people build worksheets that have date information in them. Mesa provides a quick method of filling in a range with a series of days, months, quarters, and other sequential information.

Mesa should be displaying the worksheet that you just created. Select cells A1 through A12. To select a range, move the mouse so that the arrow is on top of cell A1 (the cell in the upper, left corner of the worksheet.) Press and hold the mouse button. Move the mouse down. Notice how Mesa highlights the cells as the pointer moves over them. Move the mouse until cells A1 through A12 are selected, and then release the mouse button. You have just selected a range of cells.

Type *January* . Press and hold the Shift key on the keyboard, and then press and release ↵, and then release the Shift key. Mesa will **SmartFill** the selected cells with the sequence of months. If

you change the value in cell A1 to *Monday* and press ↵. Note that the rest of the SmartFilled range changed values. SmartFill remembers which cells were SmartFilled and automatically changes those cells when the base cell changes.

Drag & Drop Ranges

Mesa allows you to perform many tasks by direct manipulation: grabbing the thing that you want to change and changing it. Drag & Drop ranges are an example of this. You can copy of move a range of cells by dragging them to a new location.

The range A1 through A12 should still be selected from the previous SmartFill operation. Let's move this range to B2 through B13. To do this, move the mouse cursor above cell A1; press and hold the *Control* key; press and hold the mouse button. Mesa will display a shadow of the range. Move the mouse and watch as the shadow range follows the mouse movements. Move the mouse above cell B2; release the mouse button and then release the *Control* key. Mesa will move the range from A1 through A12 to B2 through B13.

Note: If you want to abort the Drag & Drop operation, release the *Control* key and Mesa will stop the operation.

You can copy a range to another location by *Alt-Dragging* it rather than *Control-Dragging* it. Mesa will make a second copy of the range in the new location.

Formula Inheritance

Mesa has a unique and useful feature called Formula Inheritance. Formula Inheritance applies object oriented concepts to spreadsheets by allowing a set of cells to inherit their formula from a master cell and automatically change the way they calculate when that master cell changes its formula.

Select cells C2 through H13 by clicking in cell C2 and dragging to cell H13 and releasing the mouse button. The range C2 through H13 should be selected and the current cell should be cell C2. Type *=INT(RAND * 100)* and press *Shift-* ↵. This will SmartFill the range with this formula that generates a random number between 0 and 99.

When a range is SmartFilled with a formula, Mesa sets the other cells in the range to inherit their formula for the formula in the base cell which is cell C2 in this case.

Select cell C2 and change the formula to *=B2 & " Hello"* and press ↵. Note that all the cells in the range changed the **way** that they calculated. Change cell C2 to *=INT(RAND * 1000)* . All the cells in the range C2 through H13 should be displaying a random number between 0 and 999.

You have just performed a SmartFill operation with formulas and taken advantage of Mesa's formula inheritance feature. Formula inheritance makes it easier to build spreadsheets because you only need to change a single formula and the other cells that inherit their formula from the base formula will automatically change the way that they calculate. Formula Inheritance also

makes spreadsheet maintenance easier because you do not have to remember all the places that the formula was copied to.

Graphing

Mesa can generate graphic representations of the number values in a worksheet. Sometimes, graphs are more effective in communicating the meaning of data than the raw data itself. Mesa supports 14 graph types and many variations within these graph types.

Select cells B2 through H13. This designates the range of data to be graphed. Select the **Graphics → Stacked Column Chart** command. Mesa will graph the data in cells B2 through H13 and present this information in a stacked column chart on your worksheet.

Mesa charts update automatically update when the data in the worksheet changes. Select the **Sheet → Recalculate** (*Command*-=) command. Note that the graph changed to reflect the new random values in the cells.

You can change the appearance or even the type of graph using the Graph Inspector. To open the Graph Inspector, first select the chart on the worksheet by clicking on it. Mesa will designate that the chart is selected by putting gray dots (handles) on the corners and sides of the chart. Open the Graph Inspector by selecting the **Graphics → Graph Inspector...** command. Mesa will display the Graph Inspector. When you de-select the chart, the Graph Inspector will disappear. It will re-appear when you select a chart again.

To change the chart type, select a new chart type from the list on the top of the Graph Inspector. To change to a 3D Bar Chart, click on the 3D Bar button on the Graph Inspector. Now change the chart to a Column Chart by clicking on the Column button.

The Graph Inspector allows you to change many attributes of a chart. The inspector is organized in a series of *Views* each of which allows you to inspect and change various attributes of the chart. By default, the Graph Inspector is open to the *Color* view. This allows you to change color attributes of the chart. To change the color or pattern of an individual element on the chart, you can use Drag & Drop color or patterns.

In the *Color* view, there is a slider underneath the *Border* color well. This slider allows you to change the width of the border around the chart. By sliding the slider to the right, you can make the border thick and by sliding the slider to the left, you can make the border thinner or disappear. Try sliding this slider to select a border thickness that you are pleased with.

Select the *Current* view in the Graph Inspector by clicking on the button in the lower left corner of the inspector marked *Current* . The Graph Inspector will display the inspector specific to the current graph type, a Column chart. The *Current* view allows you to change the width of each of the columns and to graph the first data point in each group as a line graph rather than a column. This feature allows you to graph a trend line and compare the individual column levels to the trend line. The width of the line can be changed with the slider in the inspector.

Mesa has many graph types that allow you to present your information as a graph to more effectively communicate this information. Graphs can be changed and fine tuned using the Graph Inspector.

Conclusion

Mesa contains many features to make building worksheets and presenting data intuitive and efficient. Mesa has many other features to help you work more naturally. Please take some time to read this manual so that you can discover Mesa's *Smart* features. If you have questions, please contact Athena Design's technical support team either via e-mail (feedback@athena.com) or by phone at 1.617.734.6372.

3. Getting Around Mesa

As a full-featured spreadsheet program running on the NEXTSTEP platform, Mesa allows you to do everything you expect a spreadsheet to do: financial calculations, scientific and engineering analysis, database manipulation, reporting, graphics, and more.

Basic topics

- Getting started
- Commands
- Ranges
- Inspectors
- Sizing a spreadsheet
- What' s in the window
- Moving around a worksheet
- Entering and editing data
- Working with ranges
- Formatting data
- Macros
- Saving your work

Print topics

- Quick and easy printing
- Formatting reports

Advanced topics

- Mesa preference settings
- Global style settings
- Worksheet features
- Tips and shortcuts

You can read this section from start to finish, or you can browse the topics that are of special interest to you.

If you have further questions or comments about Mesa, please call *1.617.734.MESA(6372).* Technical support is available to registered commercial users Monday through Friday from 8:30am to 5:30pm Eastern time. You can send e-mail to us at *feedback@athena.com* or send us a fax with your comments. Our fax number is 1.617.734.1130. You do not need to be a registered user to give feedback about Mesa.

Basic topics

Getting Started

To start Mesa, look in the Apps directory in your NEXTSTEP File Viewer. If you are unfamiliar with the NEXTSTEP operating environment, check the NEXTSTEP reference manual. Look for the file named *Mesa.app* or *Sunrise.app* . Double-click on the icon or the file name, and Mesa will launch. After Mesa starts, you can display a blank spreadsheet by

clicking the command **File** on the Main menu and then clicking **New** on the File submenu. This can also be done by typing *Command-n.*

Note: In order to open Mesa files, Mesa ***must*** be in your "Apps" directory or another directory checked by the W orkspace Manager , such as /LocalApps.

Opening Non-Mesa Spreadsheets.

If you have been using other spreadsheets, you may wish to open your old spreadsheets as Mesa worksheets. Mesa reads .xls files from Excel 3.0 ™, Lotus 1-2-3 ™ .wks and .wk1 files, 20/20 .m20 files, and SYLK .slk files. To open these files, launch Mesa, and use the **File → Open** command as you would for a Mesa worksheet.

Importing Non-Spreadsheet Data

To import ASCII files into Mesa, use the **File → Import Delimited Text** command. For more information about importing ASCII files, see "Import Delimited Text" on page 108.

Commands

Commands are issued to tell Mesa what you want it to do. A command is an action that you want Mesa to perform. Mesa performs the command on the current worksheet (**File → Save**), the current range (**Format → Cell Format → Bold**), or globally (**Info → Preferences...**).

Your computer's keyboard has a *Return* key (↵) in the alphabetic part of the keyboard, and an *Enter* key in the numeric keypad. These keys are **not** interchangeable in Mesa. Usually, you will use ↵ to complete an operation, but sometimes you will be instructed to press *Enter* instead.

Generally, the action taken by a command or inspector panel applies to a range that you have selected. First, select something, such as a cell or a range of cells in a worksheet, and then choose a command that acts on the selection. For example, to make the contents of a cell bold, select the cell and then select **Format** → **Cell Format** → **Bold**. If you selected a range of cells, the bold operation would be performed on the entire range. You can also perform this operation by holding down the *Command* key and typing *b* (*Command-b*).

Menus

Selecting a command from a menu is the primary way to tell Mesa to perform an action. To use the menu to issue a command, move the mouse button on top of a menu item, click down and release the mouse button. When you click down, the menu item will highlight. The item will remain highlighted until Mesa finishes performing the command.

Keyboard

Many menu commands have *Command Key Equivalents* . You can issue a command without taking your hands off the keyboard by holding down the *Command* key while typing the letter, symbol, or number that appears on the far right side of a menu command. For example, typing *Command-s* is the same

as choosing **File** → **Save** from the menu. For a list of Command Key Equivalents, see the "Keyboard Reference" on page 204.

Worksheet Tool Holders

Commands can also be placed on the Toolbar or in the Tool Holders in the upper right hand corner of the worksheet.

To place a command in a Tool Holder, first *Command-click* on any menu item. The selected command will appear as a square icon. Drag this icon to one of the squares on the worksheet Tool Holder and drop it by releasing the mouse button and the Command key. The ToolBar in the illustration above contain the **Format** → **Cell Format** → **Bold** (*Command-b*) command and the **Print...** (*Command-p*) command from the Mesa Main Menu. (See "ToolBar" on page159.)

If you decide to change the contents of either Tool Holder, simply *Command-drag* a different command into the box. The new command will replace the old one.

Note: If you place a command on top of an existing command, the existing command is over-written by the new command. If you want to blank out an item, *Command-Control-click* on the item and release the blank item on top of the item to be erased.

To make the same two commands appear automatically in any worksheet you open, use the **Windows** → **Set Current Window Tools as Default** command.

Tools may also be *Command-dragged* to any part of the worksheet.

Other Commands

There are some actions that you can perform only with the mouse using *Direct Manipulation* of the thing that you want to change. For example, to move a window around on the screen, you would click in the window's title area and drag it to the place you want it, then release the mouse button. Other examples of direct manipulation include setting colors, patterns, and fonts, resizing rows and columns, and selecting ranges. Fonts can be set using the NEXTSTEP standard font panel or by *Dragging & Dropping* font chips from a Font Well.

Scope of the Command

Each command has a *scope*. The *scope* is what the command affects or what changes the command makes. For example, **Format → Cell Format → Bold** (*Command-b*) has a scope of the current range in the current window. **File → New** (*Command-n*) affects the application by opening a new, untitled worksheet. The scope of a command does not usually change; however, some editing commands subtly change scope. For example, if you select a range of cells and issue the **Edit → Cut** (*Command-x)* command, Mesa will remove the selected range of cells from the worksheet and place them on the Pasteboard. However, if you are editing the text of a cell, select a range of characters, issue the **Edit → Cut** (*Command-x*) command, and only the characters that you have selected are cut. As you use Mesa, you will become familiar with the scope of each command. If you have used other NEXTSTEP programs, you will feel right at home with the scope of Mesa commands.

Ranges

Ranges are rectangular groups of cells. Many Mesa commands operate on ranges. The sequence of events is usually to select a range of cells and tell Mesa to perform an action on those cells. For people used to speaking English, but not used to computers, this concept may seem strange. For example, one usually forms command sentences with the verb first and then the object, e.g., “Throw the ball.” The ball is the object, the range, the thing being operated on. The verb, “throw,” is the command. In the NEXTSTEP operating environment, things are reversed. We select an object (a thing to perform an action upon) and then instruct the computer to perform the action, e.g., “The ball, throw,” “cells A1 through D10, make bold.” There is a little method to this madness. Once you select an object, you can issue many commands that affect it. For example, “Cells A1 through D10, make bold, make blue, display in scientific notation.” In English, we say, “Pick up the ball and throw it.” Once a range is selected, it becomes the “it.”

Why Ranges are Important

As outlined above, ranges serve as the “it” in the command sentences you issue to Mesa. Ranges can be formatted, moved, filled, changed, recalculated, or copied to another location or program. Ranges are the scope of data that most Mesa commands affect.

How to Select a Range

When you select a range, you tell Mesa, “This is the thing I want to work on. This is the ‘it’ in my commands. This is what’s important to me right now.”

You can select a range with the mouse or with the keyboard. We'll go through examples of both. To complete these examples, you must have a worksheet open. If you do not have a worksheet open, create a new worksheet with the **File → New** (*Command-n*) command.

There are a number of ways to select a range of cells. You can select a range using the **Sheet →Goto Cell** (*Command-g)* command. Select this command, enter the cell or range that you want in the window, then press ↵. If you want to select a range, enter the range. For example, to select cells A1 through D10, type *Command-g*. Type `A1:D10` and press ↵. Mesa will highlight cells A1 through D10.

You can also select a range of cells with the mouse. You can click in cell A1, and while holding the mouse button down drag until the mouse is over cell D10 and then release the mouse button. Mesa will select the cells A1 through D10 and place the current cell in A1. You can also click on cell A1, release the mouse button, move the mouse cursor to cell D10, and while holding the *Shift* key, click in cell D10. Once again, this will select the A1 through D10 range.

If you have never used a spreadsheet on a mouse- and window-based computer, please take a few minutes to practice selecting ranges. As most Mesa commands operate on ranges, good range-selecting skills will enhance your ability to use Mesa.

Once you have selected a range, it will remain highlighted until you select another range or cell with the mouse. You can move within the range with the *Return*, *Tab, Enter*, or arrow keys.

Inspectors

Inspectors are panels that float on the screen and allow you to *inspect* and change parts of your worksheet. There are three important inspectors: Cell Style, Sheet Format, and Graph. The Cell Style Inspector allows you to view and change attributes of a cell or ranges of cells (e.g., text color, border, display format, alignment). The Sheet Inspector allows you to view and change attributes of a whole worksheet. You can change the basic cell styles (such as default color and font), the default cell width and height, and recalculation and debugging properties. The Graph Inspector allows you to change the attributes of a graph.

On other computers, inspectors appear as *Dialog Boxes* . You invoke the particular inspector (e.g., cell color), engage in a dialog to change the color ("I'd like green, please,"), and dismiss the dialog. Mesa's inspectors are modeless. You are never in a dialog mode, but have the ability to change the attributes via the inspector at will. You can leave the inspectors open, place them out of the way, and invoke them when necessary.

Using inspectors is like shopping in a supermarket. In the old days, you had to go to the butcher to buy meat, pay for the meat, step outside, walk down the street, go into the bakery to buy bread, etc. In a supermarket, you are free to walk up and down the aisles, inspect the goods, and place the ones you like in your basket. You are never in a "butcher," "baker," "choose font," or "choose color" mode. Simply pick what you want and when you are done, check out.

For additional information and a comprehensive description of each inspector, see "Inspectors and Panels" on page 66.

Sizing a spreadsheet

When you start Mesa for the first time and open a new Mesa spreadsheet, its default size will be 32 columns by 40 rows. However, if you choose the **Info → Preferences...** command, you can set a new default size so that spreadsheets you create in the future will be any size you want them to be. (See "Preferences Panel" on page 72) The maximum spreadsheet size in Mesa is 16,384 columns by 16,384 rows.

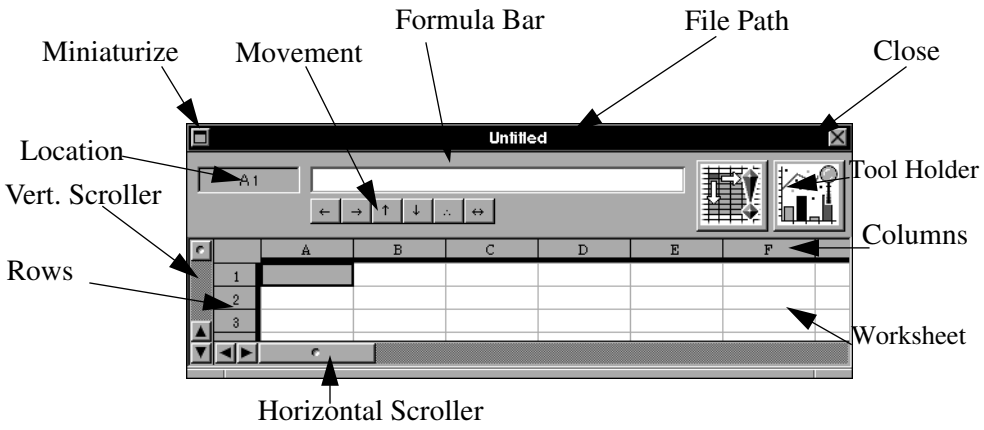
To change the size of an existing spreadsheet, select the **Format → Sheet Inspector** (*Command-Shift-G)* command to display the Sheet Inspector panel. Select the "Dimensions" view by clicking on the button marked *Dimensions* . The panel contains a Sheet Size box offering Width and Height options. Just type your selection into the fields marked "Width" and "Height". The other options on the Sheet Inspector panel are described in the Inspectors section. For more information, see "Sheet Inspector" on page 83.

What's in the window

Mesa displays spreadsheet information in a "Worksheet Window." The worksheet window contains a series of parts. Each of these parts either conveys information or performs an action.

Worksheet W indow

Part	Description
Worksheet	Displays the cells that make up a worksheet. Cells and ranges of cells can be selected by clicking and dragging the mouse on the worksheet. (See "Ranges" on page 22.)



Worksheet W indow

Part	Description
Rows	The number in the “Row Indicator” designates the row number . All cells in the worksheet are accessed by specifying the intersecting row and column of the cell. For example, the cell in the upper left corner of the worksheet is cell A1. A whole row can be selected by clicking on the corresponding row title.
Columns	The letter in the “Column Indicator” designates the column letter . All cells in the worksheet are accessed by specifying the intersecting row and column of the cell. For example, the cell in the upper left corner of the worksheet is cell A1. A whole column can be selected by clicking on the corresponding column title.
Vertical Scroller	The vertical scroller is used to move the visible portion of the worksheet.
Horizontal Scroller	The horizontal scroller is used to move the visible portion of the worksheet.

Worksheet W indow

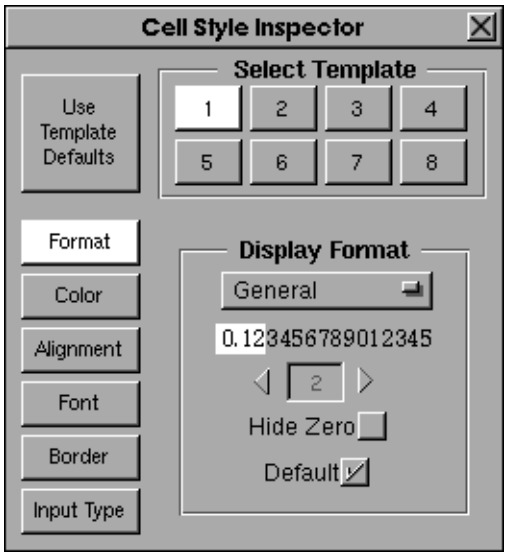
Part	Description
Miniaturize	Used to shrink the window to a mini-window. Use this to save "screen real estate."
Close	Closes the worksheet window . If this is the last window open to this worksheet, the memory used by the worksheet will be released. (See "Close" on page 109.)
File Path	The path to the worksheet file on the computer 's disk. "Untitled" if the worksheet has not been saved to disk.
Movement	Used to page the worksheet view left, right, up or down. The "triangle" moves to the home position and sets it as the active cell. (See "Moving around a worksheet" on page 23.) The "double arrow" opens the "Goto" panel. (See "Goto Cell" on page 121.)
Tool Holder	Tools are commands, reports, macros, queries, and extracts that can be placed in a window . Macros, commands, or reports that are associated with a window can be placed in the tool holder for quick access. (See "W orksheet T ool Holders" on page 20 and "The T oolBar and Tool Holders" on page 8.)
Location	The currently active cell.
Formula Bar	The area into which data and formulas are entered and edited. Direct editing in the cell on the sheet is available by selecting the **Windows → Direct Input** command. T o switch back to using the formula bar for editing, select the **Windows → Formula Bar** command.

Worksheet T ool Holders

There are two tool holders in the upper right corner of the worksheet. By default they contain tools to Save and Print, but you can change these tools for each window by placing commands, macros, reports, scripts, scenarios, or queries in the tool holders. These frames allow you to insert any two Mesa menu commands that you want, depending on your needs and work habits. Each window of a worksheet can contain a different pair of commands.

Cell style control

The command **Format → Cell Style Inspector** displays an inspector panel that allows you to control the display format of the data in any cell or cell range For a complete description, see "Cell Style Inspector" on page 75.



The inspector contains a series of "views." Each view allows you to see and manipulate different attributes that affect the display of the cell. To choose another view, select the desired view from the list in the buttons on the left side of the inspector.

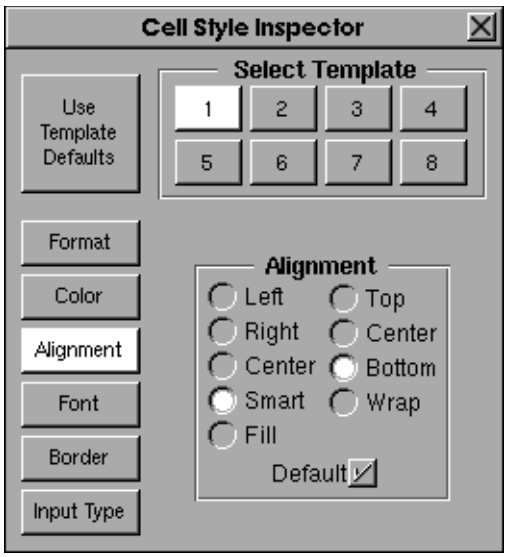
The buttons numbered 1 through 8 in the *Select Template* area display the style template for the current cell. For information about style templates, see "Style Templates" on page 76. To change the style template of the current range, click on the desired template button.

To change the display format of the current range, select *Format* and then point to the pop-up list below the caption *Display Format* and press the mouse button again. You will see an extensive list of data display formats. Move the pointer to select the appropriate format for the data you are entering into a cell or cell range. (See "Format" on page 77.)

The alignment of the text in a cell can be changed by selecting the *Alignment* view. The alignment options on the left allow you to change the horizontal alignment. The alignment options on the right allow you to change the vertical alignment. (See "Alignment" on page 79)

You can also control the data type entered in any cell or range. To do that, select *Input Type* view and select the desired input type from the list in the radio buttons. (See "Input Type" on page 83.)

The Input Type feature is especially useful if you are designing a worksheet to be filled in by others and you want to be sure that only certain types of data can be entered into certain cells. You can reserve certain cells in your worksheet for numbers, strings, dates, or formulas.



Explore the other choices on the Cell Style panel at your convenience. (See “Style Template submenu” on page 155.)

Other information on the screen

- The file name of your worksheet appears in the title bar at the top of the worksheet window. The default name is *Untitled*, which you will change the first time you save your work.
- Mesa menus appear outside the worksheet window at the left of your screen. However, you can drag any menu window to any convenient location on the screen. This is useful if you find yourself frequently using a particular submenu. For more information, see “Mesa Menus” on page 132.
- From time to time, the computer will display a small spinning-disk cursor to show that it is busy performing some function. You cannot execute any commands while the disk is on the screen. (This is the equivalent of the hourglass or wristwatch cursors you may have seen on other computer screens.)

Moving around a worksheet

Mesa provides most of the cursor controls and actions that you may have used in other products. The table below will help you learn the Mesa conventions.

Mesa Cursor Movement

Action	Mesa
Cursor to cell A1	Esc
Aborts cell edit (undo)	Command-z
Move cursor as indicated	↑ ↓ ← →, Tab, Shift-Tab
Page Right	Page Right Button
Page Left	Page Left Button
Cursor to next non-empty cell adjacent to and empty cell in indicated direction	Shift ↑ ↓ ← →
Go to specified address	Command-g, Goto Button
Page Up	Page Up Button
Page Down	Page Down Button

Entering and editing data

Entering string data

If you begin an entry with a letter, space, or punctuation mark, Mesa recognizes the entry as a string. In order to enter a number such that it will be recognized as a string, precede it with a single quote (‘) Examples of string entries include:

- Jan. 10, 1992
- Net Sales
- /USR/TOOLS/DEMO
- #3
- ’01864

Entering numeric data

Examples of recognized numeric entries include:

- 394.293
- 16-Apr -92
- 42%

Note that a date entry such as 1-10-90 will appear in the cell as *10-Jan-90* , while 7/15 will appear as *15-Jul-9* 3. If you don’t specify the year, Mesa assumes the current year. For more information on changing the formats of dates and other data, see “Cell style control” on page 28.

Entering formulas

Mesa recognizes a formula entry that begins with a number or a + or - sign. You can also start a formula with @ or =. For example, valid formulas include:

- @TODAY
- -33+58*72
- @IF(A1=3,“True”,“False”)
- =“Yes, No ”&“ Maybe”

Note: Mesa recognizes functions with and without the @ character . When using functions without @, however , you must start the formula

entry with a =, +, or - sign. For more information about entering functions and formulas, see "Functions" on page 186

Clearing data from cells

To clear data from a single cell, simply move the cursor to the cell and select **Edit → Clear**.

To clear data from one cell in a range without affecting the rest of the data in the range, move the cursor to the cell and choose the **Edit → Blank Cell** command (or simply press *Control-Shift-X* on the keyboard).

To clear a range of cells, highlight the range with the mouse and then select **Edit → Clear**. The data will be cleared, but the range will remain highlighted.

Note: Your keyboard's *Delete* key (←) deletes the character to the left of the cursor. If you press *Shift-Delete*, you can delete the character to the *right* of the cursor.

Working with ranges

Selecting a cell

To select a cell on a worksheet, simply click it with the mouse. A black border will appear around the cell and the next data you enter will appear in the cell. When you first open a worksheet, cell A1 is automatically selected.

Selecting a range of cells

To select a group of contiguous cells (a *range*), move the mouse pointer to a cell in one corner of the desired range, press the mouse button, and drag the highlight to the opposite diagonal corner of the desired range. The highlight will fill cells as you drag the mouse, so you can always tell the dimensions of the range you are defining. When you release the mouse button, the range will remain highlighted, and the cell from which you started defining the range will be selected for input.

You can then start entering data in the range. Note that as long as you use only the keyboard, the range will remain highlighted and you will be able to work with cells within the range by using the *Tab*, the ↵, and the arrow keys. If you try to move beyond the edge of your range, Mesa will wrap (either vertically or horizontally) to the next cell within the range.

However, if you click the mouse anywhere on the worksheet, the highlighting will disappear from your range. To permanently define a range, you must give it a label while it is still highlighted by selecting the **Sheet → Labels → Label Inspector** (*Command-Shift-L*) command and typing in a label name. For more information about labels, see "Labels submenu" on page 163.

Using ranges with functions

Many functions refer to data in one or more cells. In Mesa, the cursor keys can be used to specify a cell range in a function. For example:

- In the formula bar or in a cell, type the beginning of a formula such as: `@Sum(`

- Highlight a range on the worksheet with your mouse. (See "Selecting a range of cells" on page 34.)
- The range address automatically appears in the formula.
- Type the closing parenthesis to get a result like: `@Sum(F1:F4)`.
- To include more than one range in a formula, type a comma after the first range and then select another.

Mesa range descriptions

You can describe a range in Mesa in several ways. Use the one that is easiest for you. For example, you can describe the range A1 through C50 in the following ways:

- With a single period `A1.C50`
- With double periods `A1..C50`
- With an ellipsis (*Alt-period*) `A1...C50`
- With a colon `A1:C50`
- With a range *label* previously assigned to the range. (See "Labelling a range" on page 49.)

Formatting data

Display formats

Mesa supports many data display formats. The most commonly used formats are available on the **Format** → **Display Format** submenu, and most of them have keyboard shortcuts using the *Command* key.

The best way to see the various formats is to enter some numbers in a cell on your worksheet, display the **Display Format** submenu, and click each of the format options while you watch the effect on the numbers in the cell.

A more extensive list of display formats can be found via the **Format → Cell Style Inspector** (*Command-Shift-C*) command. (See "Cell style control" on page 28.)

Adjusting column width

If you enter a number that is too wide to fit in the column, you will see ########## in the cell instead of the number you typed.

To widen a column, move the mouse pointer to the right-hand edge of the column heading at the top of the worksheet until you see the pointer change its shape to two vertical parallel lines with arrows pointing right and left. When you press the mouse button you will see a box showing the column width. Drag the mouse to the right until the column is wide enough to accommodate your data.

Mesa can automatically size a column to fit all of its data. Select **Format → Row and Column Size → Smart Column Size**. You can also double-click on the column sizer (the line used to change the size of the column) to choose the best size.

If you have a range selected, you can change the size of all the columns in the range by holding down the *Command* key while sizing the column. If you *Command-double-click* on the column sizer, Mesa will choose the best column size for each of the columns in the range.

Adjusting row height

If you type some captions or other data into a row and then decide to increase the font size via the **Format → Fonts** (*Command-t*) panel, you may find that the row is no longer high enough to display the characters.

To increase the height of the row, move the mouse pointer to the bottom edge of the row at the left side of the worksheet until you see the pointer change its shape to two horizontal lines with arrows pointing up and down. When you press the mouse button, you will see a box showing the height of the row. Drag the mouse down until the row is high enough to accommodate your data.

Mesa can automatically size a row to fit all of its data. Select **Format → Row and Column Size → Smart Row Size**. You can also double-click on the row sizer (the line used to change the size of the row) to choose the best size.

If you have a range selected, you can change the size of all the rows in the range by holding down the *Command* key while sizing the row. If you *Command-double-click* on the row sizer, Mesa will choose the best row size for each of the rows in the range.

Macros

Mesa supports a full-featured scripting language called MScript. Scripts can typed directly into the MScript Inspector. For complete information about using the MScript scripting language, see “MScript” on page 306.

Saving your work

Mesa offers three options for saving your work.

- Use **File** → **Save** (*Command-s*) to save a worksheet and continue working on it.
- Use **File** → **Save As** (*Command-Shift-S*) to save a worksheet under a new name.
- Use **File** → **Close** (*Command-Shift-W)* to close the worksheet. If you have not saved the worksheet, Mesa will ask you if you want to save the worksheet before closing it.

For more information about these and other file options, see "The File Menu" on page 139.

Print topics

The **Print...** command on the Main Menu displays the Print dialog panel. You can elect to print a named report, a selected range from your worksheet, or the entire worksheet. (See "Print" on page 134.)

Since printing your work is probably an important part of your job, you will want to know how to do the following things:

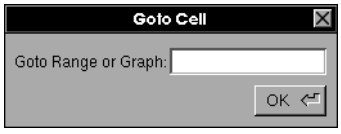
Quick and easy printing

Suppose you have entered the following table into Mesa and would like to print it without printing the whole worksheet.

Before issuing the Print command, highlight the range you want to print so that it looks like this:

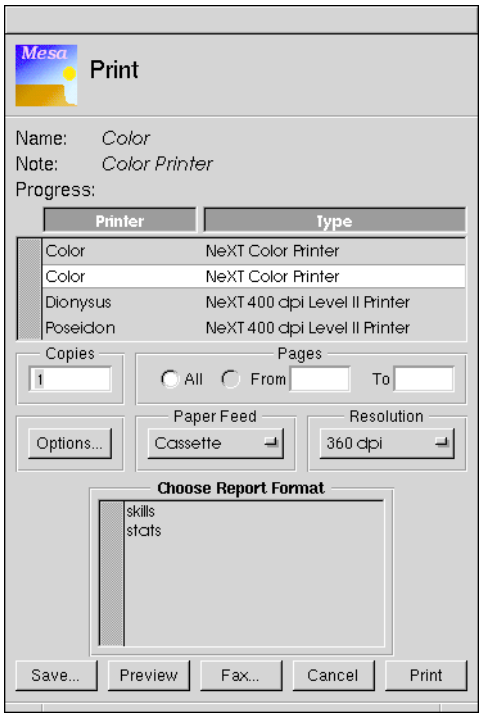
	A	B	C	D	E	F	G
1		Sales Commissions					
2		Q3 '91	Q4 '91	Q1 '92	Q2 '92	Q3 '92	Totals
3	John	$20,655	$16,829	$97,426	$24,521	$79,785	$239,216
4	Mary	$27,822	$17,906	$71,739	$84,709	$16,227	$218,403
5	Sue	$92,824	$43,155	$74,071	$36,182	$50,381	$296,613
6	Ralph	$31,678	$2,862	$48,672	$20,543	$53,676	$157,431
7	Lou	$68,606	$31,120	$42,652	$53,387	$48,883	$244,648
8	Dana	$39,865	$91,122	$24,898	$70,551	$31,469	$257,905
9	Total	$281,450	$202,994	$359,458	$289,893	$280,421	

If your range extends beyond the edges of the worksheet, then you can use the **Sheet → Goto Cell** panel (*Command-g*) to define the range to print.



For example, type the range `A1:Z200` in the *Goto Cell* box, click *OK*, and Mesa will select the range from A1 to Z200 to print.

Now you are ready to print. Click the **Print...** command on the Main Menu (*Command-p*) to display the Print dialog panel.



If you examine the dialog panel you will see the following default settings:

- The output will be sent to your local default printer
- All pages of your selected range will be printed
- One copy will be printed

If these defaults are acceptable, just click the *Print* button and the output will be printed.

Paper size and orientation can be changed via the **Format** → **Page Layout...** (*Command-Shift-P*) command. Mesa will display the Page Layout dialog box

which allows you to change print parameters, add headers and footers on each page, adjust margins, and optionally display row and column titles. For more information, see “Page Layout...” on page 156. The information in Page Layout will not affect Report Formatting. For more information about reports, see “Formatting Reports” on page 41.

Formatting Reports

In Mesa, more complex reports can be created using the Report Layout Inspector. Select **Format → Report Layout...** (*Command-Shift-R)* to display the Report Layout Inspector that lets you perform detailed report formatting.

A detailed description of the Report Format Inspector is given in “Report Layout Inspector” on page 102.

Page breaks

In Mesa, each range automatically prints on a separate page. For information about creating additional page breaks, see “Page Breaks submenu” on page 156.

Creating and printing headers and footers

Mesa uses report *Tags* to let you put special headers anywhere on a page. Tags print the specified information on each page. They also allow you to place items of information (text, page numbers, dates, etc.) anywhere on the page. Tags are available for cell contents, the current date or time, the filename, or the pathname.

You can also create custom tags containing other information. For information about creating custom tags, see “Creating and printing custom tags” on page 111.

Mesa offers formatting symbols that may be positioned anywhere on the page.

Mesa T ag Formatting Codes

Format	Mesa Code
Page number	&p
Total # of pages	&n
Current date	&d
Current time	&t
Document name	&f
“&”	&&
Cell contents	Use the “Cell Contents” tag

Text in tags is always left-aligned. To change the position of the tag, simply drag it to where you want on the page.

Advanced topics

Mesa includes many useful features that will make your work easier. Here is a sampling of the ones you may want to use in your day to day work.

Mesa preference settings

You can customize certain aspects of Mesa to suit your own needs and preferences. Do this by selecting the **Info** → **Preferences...** command to display the Preferences dialog panel. For more information, see "Preferences Panel" on page 72.

The Preferences panel lets you set such things as the default worksheet size, the date and time formats, the currency symbol, select AddIns (AddIns are not available in Sunrise), set auto-execution of macros, and so forth. Changes you make on the Preferences panel are automatically saved when you click the *OK* button.

Global style settings

- The **Format** → **Sheet Inspector** command (*Command-Shift-G*) displays an inspector panel that gives you great control over the content and appearance of your worksheets. For complete information, see "Sheet Inspector" on page 83. You can override the global worksheet style for any cell or range by using the Cell Style Inspector . For complete information, see "Cell Style Inspector" on page 75.

Highlights of the Sheet Inspector Panel include:

- The *Format* view with its pop-up list of number formats and a decimal slider that lets you select from 0 to 15 decimal places.
- The *Alignment* view that lets you align data in your cells in several ways, both vertically and horizontally .
- The *Cell Size* sliders that let you adjust the default height and width of the cells on the worksheet.

Worksheet features

Non-scrolling titles

You can "freeze" a row or a column of titles so that they stay in view while you scroll the worksheet up and down or right and left. To make the top row of your worksheet non-scrolling:

- Be sure the row of titles is visible on the screen.
- Move the mouse pointer to the heavy black line below the column headings until you see the pointer change its shape to a down arrow .
- Press the mouse button and drag the line down until it is below the title row you wish to freeze.
- Release the mouse button.

You can do the same thing with titles in a column by using the heavy black line to the right of the row numbers. And you can always reverse the process by dragging the border line back to its original position.

Multiple windows

Mesa allows you to open multiple (two or more) windows of the current worksheet so you can see as many different parts of your work as you need to.

For example, to display rows A1..A20 of your worksheet in one window and rows A200..A225 in another window:

- Select the command **Windows → New Window** (*Command-Shift-N*) and a new window will appear on your screen.
- Your original worksheet name will now show the suffix ***…1*** and the new window will have the same worksheet name with the suffix ***…2***.

- Use the scroll bar to bring row A200 into view in the new window .

After you create one or more new windows, you can move among them by simply clicking the one you want to use. You can display the **Windows** menu, which lists the names of all windows below the rest of the menu selections. If a window is obscured by other windows, you can bring it to the front by clicking on its name in the **Windows** menu.

Automatic recalculation

When you create a new worksheet, *Auto Recalc* is on by default. Every time you change any data, the worksheet is recalculated automatically.

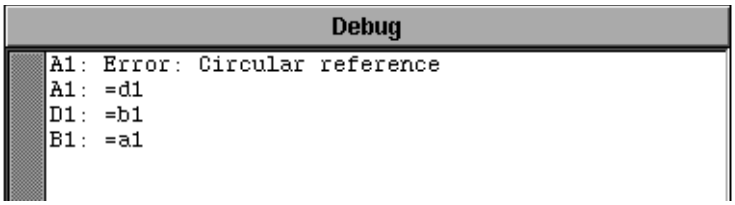
To turn off this feature, select **Format → Sheet Inspector** (*Command-Shift-G*), select the *Recalc* view, and click the *Auto Recalc* box to remove the check mark. Then, when you want to recalculate your worksheet, select **Sheet → Recalculate** (*Command-=*).

Circular references

Mesa has a very helpful way of showing all cells involved in a circular reference. To see how this works:

- Select **Format → Sheet Inspector** (*Command-Shift-G*), select the "Recalc" view , and click the *Debug* button to see the pop-up list. Select *Stop on Circ.*
- In the worksheet enter the following values:
  - in A1 enter: *=d1*
  - in B1 enter: *=a1*
  - in D1 enter: *=b1*
- Press *Command-=* to recalculate the worksheet.

- You will see a Debug panel identifying the first cell in the circular reference and all the other cells involved.



Copying cell data

You can copy cell data from one range to another in two ways: Copying the data to the Pasteboard or Dragging the range.

The first step is to choose the range of cells that you want to copy. See "Selecting a range of cells" on page 34.

To copy the range via the Pasteboard, select **Edit** → **Copy** (*Command-c*). This will copy the selected range of cells to the Pasteboard. Mesa saves data on the Pasteboard in a number of different formats (these formats are selectable from the Preferences Dialog) including an internal format, ASCII, and EPS. You can copy a range of cells and paste the graphic image of these cells into a word processing or presentation program. The image that you paste will contain the complete formatting information that it contained on the spreadsheet. If you want to copy the range of cells to another place on the current Mesa worksheet or to another worksheet, select the upper left cell in the destination (the place you want to copy the cells to) and select **Edit** → **Paste** (*Command-v*). Mesa will place the copied range in the new location and adjust the relative cell references in the formulas to reflect the new location.

You can also directly *Drag & Drop* a range of cells. Once you have selected the range to be copied, move to the area on the sheet where you want to copy the range to. Press and hold the *Alternate* key and click down the mouse button. Mesa will show the source range on top of the current area. You can drag this image around until you are satisfied with the position. Release the mouse button and Mesa will copy the range and highlight the new range.

Note: If you want to abort the copy operation, release the *Alternate* key before you release the mouse button. This signals Mesa that you want to stop the copy operation.

Moving cell data

You can move cell data from one range to another in two ways: Moving the data via the Pasteboard or Dragging the range.

The first step is to choose the range of cells that you want to copy. See "Selecting a range of cells" on page 34.

To move the range via the Pasteboard, select **Edit** → **Copy** (*Command-c*). This will copy the selected range of cells to the Pasteboard. You can move data from only one place to another on the same worksheet. Select the upper left corner of the location that you want to move the range to. Select **Edit** → **Paste Special** → **Move**. Mesa will move the copied range to the

new location on the worksheet and update formulas that referred to the old locations so that they refer to the new location.

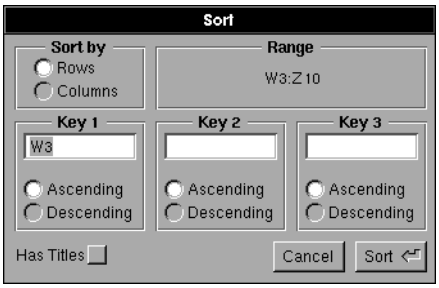
Note: The Move operation is different from a Cut and Paste operation. If you Cut and Paste a range of cells, formulas that referred to the old range continue to refer that range. If you Copy and **Paste Special** → **Move** the cells, formulas in your worksheet are updated to reflect the new location of the range.

You can also directly *Drag & Drop* a range of cells. Once you have selected the range to be moved, move to the area on the sheet where you want to move the range to. Press and hold the *Control* key and click down the mouse button. Mesa will show the source range on top of the current area. You can drag this image around until you are satisfied with the position. Release the mouse button and Mesa will move the range and highlight the new range.

Note: If you want to abort the move operation, release the *Control* key before you release the mouse button. This signals Mesa that you want to stop the move operation.

Sorting data

Highlight the range you want to sort and then select **Sheet** → **Database** → **Sort** to display the Sort dialog panel shown below.



The range you selected will be shown in the *Range* box. Select your sort options and click the *Sort* box. (If the first row or column of your range contains titles, click the *Has T itles* box to prevent the titles from being sorted.)

Labelling a range

If you enter a name in a cell, you can assign it as a range label to any adjacent cell by using the **Sheet → Labels** menu. For example, if you enter *Sales* into a cell and select **Sheet → Labels → Label Range to Right**, the label Sales will now refer to the adjacent cell on the right. You can also choose *Left, T op,* and *Bottom* . (See “Labels submenu” on page 163.)

Displaying your range table

After you have assigned a number of range labels in your worksheet, it may be helpful to be able to display a list of them for review. You can do this by highlighting an unused cell in a blank area on your worksheet and selecting **Sheet → Labels → Insert Label Index**. A list of the names and ranges will be displayed, starting in the cell you highlighted. (See “Labels submenu” on page 163.)

Converting formulas to values

To convert the formulas in a range to constant values, highlight the range and then select **Edit → Convert → Formulas to Values**. This freezes the values. They will no longer change when the worksheet is recalculated.

Protecting ranges

Within your worksheet, you can designate protected and unprotected cells.

When you create a new worksheet, protection is off by default. After you have built a structure that requires some ranges to be protected and others open for data entry, you should do the following:

- Select **Format → Sheet Inspector** (*Command-Shift-G*), select the "Protection" view , and click the *Protection* button to display the check mark. The entire worksheet is now protected.
- Now, highlight a range you want to unprotect and select **Format → Cell Style Inspector** (*Command-Shift-C*) and select the "Input T ype" view .
- Select the *Unprot'd* radio button.
- The range you selected is now unprotected.

Importing files

You can import a file containing text with various types of delimiters into your worksheet. To import a file:

- Highlight the cell in the upper left corner of the area where the imported text will go.
- Select **File → Import Delimited Text...** to display a dialog panel.
- Choose the appropriate input specifications.

- Choose the file to import and click the *OK* button.
- The contents of the file will appear in your worksheet.

Linking Worksheets

You can link Mesa worksheets so that information from one worksheet can be used in another. The file link is entered into a cell, just like a formula.

Mesa's file link syntax is:

- filename::range
- or
- filename!range

where *filename* is the path to the file to be linked. If *filename* is a path, it must be an absolute path. If it is not a path, Mesa searches for the file in the directory where current worksheet is saved. *Filename* can either be a string or a cell reference. This allows you to link to different worksheets based on the contents of a cell *Range* can be a cell address, range, or label.

If both files are in the same directory, you can use just the filename. Otherwise, you must specify the path of the linked file. Examples are given in the table below.

Linked Worksheet Examples

Formula	Cell X1	Result
="Sales"::A1	Prices	Returns contents of cell A1 in worksheet "Sales" located in the same directory as the current worksheet.
="/sam/Sales.Mesa"::c5	Prices	Returns the contents of cell C5 in the "/sam/Sales" worksheet.

Linked W orksheet Examples

Formula	Cell X1	Result
=sum("Sales"!a1:b7)	Prices	Returns the sum of cells A1 through B7 in the "Sales" worksheet located in the same directory as the current worksheet.
=sum("Sales"::W estern)	Prices	Returns the sum of the cells in the range "W estern" in the "Sales" worksheet located in the same directory as the current worksheet.
=X1::A1	Prices	Returns the contents of cell A1 in the worksheet "Prices" located in the same directory as the current worksheet.
=X1::A1	Sales	Returns the contents of cell A1 in the worksheet "Sales" located in the same directory as the current worksheet.
=sum(X1::widgets)	Prices	Returns the sum of the "widgets" range in the "Prices" worksheet located in the same directory as the current worksheet.
=max(X1::labor)	/sam/LaborCost	returns the largest value in the "labor" range in the "/sam/LaborCost" worksheet.

Note that the NEXTSTEP environment is sensitive to capital letters in file names, so be sure to spell file names exactly as you see them. File names may contain spaces, which must be included in link references as well. When you work with linked files, be aware of the following points:

- When you open a worksheet that has links, recalculate using *Command*-= unless Auto Recalc is on.

- Links can be used in formulas, just like any other valid expression. For example: *@PV(2500,4%, "US_SALES"::b5*).

Referencing files via cells (i.e., *Q5::Prices*) is a powerful tool that allows changing where the data for a calculation is coming from without changing the formula. If you have sales data from four regions saved in worksheets named, *North* , *South* , *East* , and *West*, you could build a summary worksheet to generate totals for each region. The formula *=sum(b5::total_sales)* would sum the total_sales region for the sheet named in cell B5. To see the summary for *North* , enter "North" into cell B5 and select **Sheet** → **Recalculate** (*Command-=*). To see the summary for *West*, enter "West" into cell B5 and select **Sheet** → **Recalculate**.

Tips and shortcuts

This section lists several shortcuts that are available in Mesa.

In-cell editing

The Formula Bar is the default mode for data entry in a Mesa worksheet. (See "What's in the window" on page 25.) However, you can enter data directly into a cell by selecting **Windows** → **Direct Input**. To change back, select **Windows** → **Formula Bar**.

Customizing "home"

Normally, the *Esc* key sets the current cell to the upper left corner of the worksheet. If you want to reassign the home cell so that pressing the "home" key will bring you to that new location, highlight the new home cell and press *Control-Esc* . Now, when you

press the *Esc* key, the highlight will go to the cell you chose. You can also go "home" by clicking on the ∴ button on the worksheet window.

Mesa allows you to open more than one window to the same worksheet. The home may be set independently in each window, and each window remembers its home when you save the file. This allows you to have different views into a very large worksheet and customize each view so that it knows its own home location.

Selecting large ranges

If your range extends beyond the edges of the worksheet, use the **Sheet → Goto Cell** (*Command-g*) dialog panel to define the range. (See "Quick and easy printing" on page 38.)

Selecting rows and columns

To select an entire row or column, just click the mouse on the row or column heading. If you *drag* the mouse pointer across the headings, you will select several adjacent rows or columns.

Selecting all cells

To select all cells at the same time, click the empty box in the upper left corner of the worksheet (just above row 1 and to the left of column A).

You can also select all the cells on the sheet using **Edit → Select All** *(Command-a)*.

Smart-sizing rows and columns

To easily resize a row or column to accommodate long entries or large fonts, use the **Format → Row and Column Size** menu. First, highlight the rows or columns to be resized, then display the menu and select the option you want. (See "Row and Column Size submenu" on page 154.)

You can also *Smart-size* a row or column by double-clicking on the row or column sizer. You can *Smart-size* a range of rows or columns by highlighting the range and *Command-double-clicking* a row or column sizer in the range.

Smartsum

To quickly create a total for a row or column of numbers, highlight the row or column you want, plus an empty cell at the end. Then select **Sheet → Smart → Smartsum** or press *Command-Shift-&*. Mesa will place a SUM() formula in the empty cell where the total will appear. You can use the other selections in the **Sheet → Smart** submenu to perform quick statistical analysis on ranges.

AutoGrow ranges

This feature, applied to a named range, allows the range to grow beyond its defined size when new information is added. To use AutoGrow in a range, try this:

- In your worksheet, highlight a range such as A1:D5.
- Select **Sheet → Labels → Label Inspector…** (*Command-Shift-L*) to display the Label Inspector Panel.

- You will see your range address in the box at the lower right of the panel and the text cursor blinking in the range name box at the left.
- Type in a range name, click the *AutoGrow* box to display the check mark, and click the *Add* button to add the range to the label list.
- Close the inspector panel and return to the range, which is still highlighted.
- Use the ↵ or ↓ key to move the cell cursor past the bottom row of the range and see that a new row is added to the range each time you press the key . The formula and formatting information is copied from the line above.

AutoGrow ranges are useful for data entry templates, because you don't need to pre-allocate and format a range for data entry. Set the range to AutoGrow and as more data entry space is needed, the range will expand automatically.

Constrained ranges

When you enter a formula containing a range label in your worksheet, Mesa lets you limit (constrain) its effect to specific rows and/or columns of a labeled range. Let's assume you have a range called REGION (A1:D200) and you want to add up the data in column C.

You can enter a formula as `@SUM(REGION[][2])`, meaning "add up all rows in the third column of the named region." In this case, Mesa row and column numbering begins with 0.

The syntax is: *function* (*range label* [*rows*][*columns*])

Study the examples in the table below, keeping in mind that, if *all* columns in the range are to be included, the second set of brackets can be omitted. For example:

- @SUM(REGION[0])
- @SUM(REGION[][1,2])
- @SUM(REGION[4,9])
- @SUM(REGION[29,73][1,2])

The items in the brackets can be either numbers or the following keywords:

Keyword	Action
THIS	Constrains the range to the current row or column
NEXT	Constrains the range to the next row or column
PREV	Constrains the range to the previous row or column
BEGIN	Constrains the range to the first row or column
END	Constrains the range to the last row or column

The above keywords can be mixed with each other and absolute row or column numbers. For example, SUM(SALES[BEGIN,THIS]) computes the sum of the range SALES from the first row through the current row.

SAME(), @SAME()

You can enter a formula in one cell and have a range of cells share the same formula so that changing the formula in the original cell changes it in the related range as well.

Formula inheritance is one of Mesa's most powerful features. It has two major benefits: making sheets easier to maintain, and simplifying complex commands such as IF() and @IF() functions.

The SAME() function is like a copy of the formula at the time the worksheet is recalculated. For example, if you have the formula =A1+1 in cell B1 and the formula =SAME(B1) in cell B2, the formula in cell B2 will be evaluated as if it contained =A2+1. This is as if you had copied the formula from cell B1 into cell B2 when you recalculated the spreadsheet. If you make a change in the formula in cell B1, cell B2 will change ***how*** it calculates its value.

NEXT(), @NEXT()

This function allows you to generate sequences of months, days, dates, and so forth. It is a very useful tool when you are creating column or row headings.

Style T emplates

Mesa lets you define up to eight different worksheet display formats or *style templates* . Once defined, style templates can be applied to any cell or range in your worksheet.

Style templates are useful for controlling the appearance or content of different ranges in the same worksheet. For example, you can apply individual style

templates to titles, income data, expense data, profits, losses, and so forth, to make these parts of the worksheet stand out.

The default format of your worksheet is Style Template 1, which is defined and changed on the Sheet Inspector Panel (select **Format → Sheet Inspector** or *Command-Shift-G*). For more information, see"Style Templates" on page 76 and "Sheet Inspector" on page 83.

4. Tutorial

Introduction

This chapter is a short tutorial that will step you through building a simple check-book worksheet. This worksheet will take advantage of many of Mesa's features including AutoGrow ranges, predefined Reports, formulas, formatting, data entry validation, and other Mesa features.

There are six lessons in this chapter. The resulting worksheet at the end of each lesson is located in the *MesaSampler/Tutorial* directory. This will allow you to compare your work to the result of the lesson. It also lets you skip lessons that cover material that you are already familiar with.

Creating the Worksheet

The first step is to create a new worksheet. A worksheet is the place that you work on in Mesa. It is a collection of cells organized in rows an columns. Each cell can be individually addressed with its row and column address. Rows are numbered 1 through 16384 and columns are labeled A through XFD. When you first create a worksheet, it will have 32 columns and 40 rows. It will automatically expand as you need more space. You can also make a worksheet larger or smaller by changing the sheet size in the Sheet Inspector.

To create a new worksheet, select the **File → New** (*Command-n*) command. To do this, move the mouse cursor above the **File** selection on the main menu. Click and release the mouse button. The **File** submenu will appear. Move the mouse cursor above the **New** selection, click and release the mouse button. Mesa will create a new, blank worksheet.

Many Mesa commands have *Command-Key* equivalents. These commands can be issued by clicking on the desired menu item or by pressing the *Command* key along with another key on the keyboard. Command-Key equivalents are useful shortcuts once you become familiar with an application. The Command-Key equivalent will be listed after each command so that they will become familiar.

Once you have created a new, blank worksheet, you can save it on your computer's disk. Saving a worksheet transfers the information contained in the worksheet from the computer's internal memory to a hard disk or file server. Once the information is on *permanent* storage, it will remain until it is changed or erased.

It is a good idea to save your work from time to time in case there is an unexpected problem with your computer. If you have saved the file, even if the information in the computer's memory is lost, you will still have the information on disk.

To save a worksheet, select the **File → Save** (*Command-s*) command. The first time you save a worksheet, Mesa asks you for the name of the worksheet. Select an appropriate name, for example, *Checking*, enter it into the name field in the Save panel and press ↵.

Entering the Column Headings

Now that you have created the worksheet and saved it, you can put information into the worksheet. This section will show you how to put labels on the top of column of data and then *freezing* these titles so they remain on the worksheet even if you scroll to a different section on the worksheet.

The current cell should be cell A1. If it is not, move the mouse cursor on top of cell A1 and press and release the mouse button. Enter *Date* and press the *Tab* key. The string *Date* will appear in cell A1 and Mesa will change the current cell to B1. Enter *Deposit* into cell B1, *Withdrawl* into cell C1, *Reason* into cell D1, *In Book* into cell E1, *Balance* into cell F1, and *Off By* into cell G1. You have entered the text of the titles.

Next, you will put underlines beneath each of the titles to set them off. Select cells A1 through G1. To do this, position the mouse cursor over cell A1. Press and hold the mouse button. Drag the mouse to the right. Notice that Mesa is highlighting each of the cells under the mouse. Move the mouse until cells A1 through G1 are highlighted. Release the mouse button. You have selected cells A1 through G1.

Open the Cell Style Inspector by issuing the **Format → Cell Style Inspector...** (*Command-Shift-C*) command. The Cell Style Inspector allows you to examine and change styling elements for the cells in the range that you have selected. The Cell Style Inspector contains a series of views that allow you to view different aspects of the cell's style. Click on the button marked *Font* on the left

side of the inspector. This view allows you to change the font and also change the underlining of the font in the cells. Click on the radio button marked *Wide* . This will place a wide underline beneath the text in the selected cells. Note the change on the screen.

The next step is to make the width of the column containing *Reason* a little wider. Move the mouse so that the mouse cursor is positioned over the line between the column titles *D* and *E*. Note that the cursor changes from an arrow to a vertical bar with two smaller arrows pointing to the left and right. Press and hold the mouse button. Note that Mesa displays the width of the column. Drag the mouse to the right until the column is 1.75 inches wide and release the mouse button. You have made the column wider.

The final step is to freeze the titles so that they will remain at the top of the worksheet even as you scroll through the data on the worksheet. Position the mouse cursor on top of the thick black line that is just beneath the column labels. Note that the cursor has changed to a horizontal bar with a single down arrow. Press and hold the mouse button. Drag the mouse down until the dark grey line is below row 1. Release the mouse button. Mesa has frozen the first row at the top of the worksheet.

Move to cell A2 and save the worksheet by issuing the **File** → **Save** (*Command-s*) command. You have completed the second lesson in the tutorial. Your worksheet should look like *Lesson2.Mesa* in the *Tutorial* folder.

Entering the Formulas

This lesson will teach you about entering formulas and creating labels on the worksheet. Formulas are algebraic statements that can be evaluated by Mesa to return a value in a cell. The formula in one cell can refer to the value in another cell. When the value in a cell changes, the spreadsheet automatically recalculates and updates each cell that relies on a value that was changed.

The first thing that we will do is create a label or a named range called *Checking* . A label can be referred to in a formula by its title rather than the cells addresses. Referring to a label makes formulas more readable. Select cells A2 through G3. Open the Label Inspector by issuing the **Sheet** → **Labels** → **Label Inspector...** (*Command-Shift-L*) command. Type *checking* and press ↵. Mesa will create a label called *checking* and assign it to cells A2 through G3.

The next step is to enter the formulas that calculate the balance in the checking account. Move to cell F2 and enter the formula *=B2-C2* . This formula subtracts the amount withdrawn from the amount deposited and places the result in cell F2. Next, move to cell F3, and enter the formula *=F2 + B3 - C3* . This formula added the current transaction to the prior balance, thus accumulating the balance in the checking account.

The next step is to sum the total amount deposited and withdrawn. Select cells B4 and C4. Type *=SUM(checking[][this])* and press *Shift* -↵. This will SmartFill this formula into the selected range. The formula sums, adds up, all the values

in the range *checking*, but constrains the range to *this* column. By SmartFilling the range, this formula is automatically propagated into cell C4.

The next step is to enter the formula into the column that alerts you if there is a difference between the amount in the checkbook and the amount that should be in the checkbook. Select cells G2 and G3. Enter the formula *=setcolor(setbkgcolor(if(#NOT#isempty(e2),e2-f2),if(if(#NOT#isempty(e2),e2-f2)<>0,redcolor, whitecolor)),if(if(#NOT#isempty(e2),e2-f2)<>0,whitecolor,blackcolor))* and press *Shift-* ↵. This formula sets the foreground color to white and the background to red if the value in the checkbook is different from the running total.

You have entered all the formulas into the worksheet. Please save your work. Your worksheet should be similar to *Lesson3.Mesa* worksheet.

Formatting

This lesson will demonstrate some of Mesa's formatting capabilities. Formatting changes the visually appearance of the worksheet to make it more pleasing and more effective at communicating information. Mesa allows you to change the format by direct manipulation, via menu commands, and via inspector panels. This lesson has examples of using the menu items and inspectors to change formatting.

Select cells E2 through G3. Select the **Format** → **Display Format** → **Currency** (*Command-Shift-$*) command. This makes all the cells in the range display their numbers in currency format with

two decimal places. Select cells B2 through C3 and apply the currency format to those cells as well.

Select cells B4 and C4. Open the Cell Style Inspector using the **Format → Cell Style Inspector...** (*Command-Shift-C*) command. Change the display format from *General* to *Currency* using the pop-up list in the middle of the inspector. You have just changed the formatting of these cells the same way you would have by using the **Format → Display Format → Currency** (*Command-Shift-$*) command. Next, change the view in the Cell Style Inspector to the *Border* view by clicking on the button marked *Border* on the left side of the inspector. This will allow you to inspect the border attributes of the selected range. There are four check-boxes organized in a diamond shape in the inspector. These boxes control the border around the cell. Click on the top-center check-box. This will place a thin border on top of the selected range.

Mesa allows you to set the type of input that a cell will accept. This is valuable in order to make sure that data entry is done correctly. By default, Mesa tries to figure out the type of information that is entered into a cell (i.e., number, date, formula, etc.) You can control the type of information entered into a cell by defining the cell's Input Type using the Cell Style Inspector.

Select cells A2 and A3. Change the view on the Cell Style Inspector to *Input Type*. Change the input type to *Date* by clicking on the date radio button. You have designated this cell as date only. Note that the cell type designation *Date* is in light gray in below the cell number indicator in the upper left corner of the worksheet. Set cells

B2 through C3 to *Number* format. Set cells D2 and D3 to *String* format. Set cells E2 and E3 to *Number* format. Finally set cells F2 through G3 to *Formula* format.

You have entered the formatting for this worksheet. You have changed the display format of number cells to display their contents as currency. You have put borders on cells and set the input type of cells. Please save your worksheet. It should look like the *Lesson4.Mesa* worksheet.

AutoGrow Ranges

This lesson will show you how to create an AutoGrow range and do direct data entry. An AutoGrow range is a data entry range that automatically grows when you need to enter more data into it. This lesson will also demonstrate how to do direct data entry rather than using the formula bar at the top of the worksheet window.

First, move to cell A2. Open the Label Inspector by issuing the **Sheet → Labels → Label Inspector...** (*Command-Shift-L*) command. Select the label named *Checking* by clicking on it in the browser. Click on the check-box marked *AutoGrow*. You have just designated this range as an AutoGrow range. Select the *Checking* label and click on the button marked *Goto*. This will select the *checking* range on the worksheet.

Issue the **Windows → Direct Input** command. This command allows you to enter and edit data directly in the cell, not in the larger formula bar in the top of the worksheet. Direct Input is useful when doing data entry because you can watch the

data that you are entering rather than shifting your eyes between the formula bar and the place that you are doing the data entry in the worksheet.

This lesson has taught you how to mark a range as AutoGrow, how to select a range from the Label Inspector, and how to select Direct Input rather than Formula Bar input. Please save your worksheet. Your worksheet should look similar to the worksheet in *Lesson5.Mesa* .

Reports

This lesson will show you how to build a simple report and how to print it.

Open the Report Inspector by issuing the **Format → Report Layout...** (*Command-Shift-R*) command. Mesa will display the Report Inspector. The Report Inspector allows you to name and define a report that can be printed with the push of a button.

First, we will name the report. Click in the *Name* field in the lower right part of the Report Inspector and type *Checking* and click on the *Add* button. This will add a report named *Checking* to the worksheet.

Next, change the orientation of the report by clicking on the *Landscape* radio button in the *Page Layout* section of the inspector. Select the first line in the *Range Information* section and type *A2:G999 T=A1:G1* , and click on the *Smart Width* button on that line. This will set the print range to cells A2 through G999 and use the first

row as titles on each page. You may question the designation of almost 1,000 lines in the report if you have only filled in a few lines. Mesa automatically ignores rows that do not contain any information when printing, so only the lines that contain information will be printed.

Now, we will add *Tags* to the report. The Report Inspector contains a *virtual piece of paper* onto which you can place *tags* which contain information such as print time and more. You can reduce the scale of the piece of paper using the *View Scale* slider on the bottom of the Report Inspector. Slide the slider until you can see the whole piece of paper (about 40%). Drag the tag marked *File Name* from the tags and drop in on the top part of the piece of paper. Mesa will now print the file name on the top of this report every time you print it.

We will now add a print button to the worksheet so that you have easy access to printing this report. Press and hold the *Command* key. Click on the button on the Report Inspector marked *Print: checking* . Drag the button and drop it onto one of the two tool holders in the upper right corner of your worksheet. Every time you click on this button, Mesa will print out the checking report.

This lesson has shown you how to add a report to a worksheet. Please save this worksheet. It should be similar to the *Lesson6.Mesa* worksheet.

Summary

This has been a short tutorial that has stepped you through building a worksheet. You created a new worksheet, entered information and formulas, formatted the information, and created a report. Mesa has many other features that will help you enter data and present it in a pleasant and informative fashion. If you have any questions about this tutorial or about using Mesa in general, please contact Athena Design Technical Support at 1.617.734.6372 or by e-mail at feedback@athena.com.

5. Inspectors and Panels

Inspectors are tools used to view and change parameters and items within the worksheet. The data and formulas are entered directly into cells in the worksheet; however, formatting cannot be done as easily. Inspector panels float along side the worksheet window. They offer views of various different settings of cells, the whole worksheet, reports, etc. For example, the number of decimal places to be displayed is shown and can be modified via the “Format” view of the Cell Style Inspector.

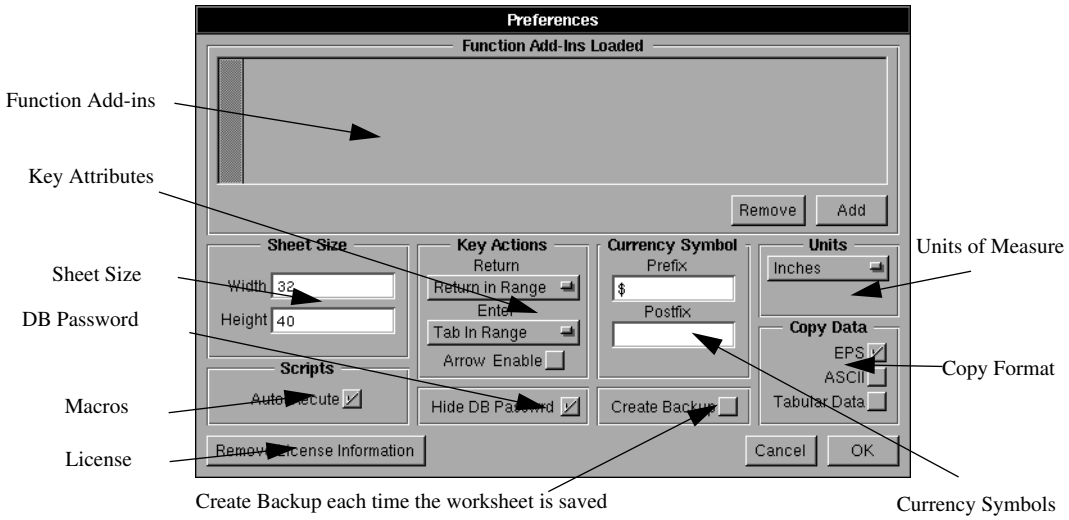
To make inspectors as small as possible, some inspectors have multiple views. A view contains a subset of the tools in the inspector. For example, the Cell Style Inspector contains views to manipulate display format, cell color, cell contents alignment, font, cell border and border color, and cell input type. A list of available views for an inspector appears in a column of buttons on the left side of the inspector. To change the inspector view, click on the button corresponding to the desired view. Changing views does not alter data or formatting information, so you can explore the different views in each inspector without changing anything in the worksheet. Some inspectors have only one view.

Inspectors are associated with the current worksheet. The current worksheet is designated by the black or dark gray title at the top. If the current worksheet changes, the inspectors are updated to reflect the contents of the new worksheet. If there is

nothing to inspect, inspectors will automatically close. If there is no active worksheet, the Cell Style, Sheet, Report Layout and Label Inspectors will close. If there is no chart selected on the current sheet, the Graph Inspector will close, but it will reopen automatically when a chart is selected.

If you are accustomed to using a Macintosh or a PC running Windows, you may be familiar with dialog boxes. Dialog boxes are modal versions of inspectors. When they are open, the program is in the mode to have a dialog only about the topic related to the dialog. Inspectors are more flexible in that the worksheet can be accessed when an inspector is open.

Preferences Panel



The Preferences dialog panel lets you set the default spreadsheet size of any new spreadsheets you create by entering a default width and height in the Sheet Size box. Selecting **Info→Preferences** opens the Pref-

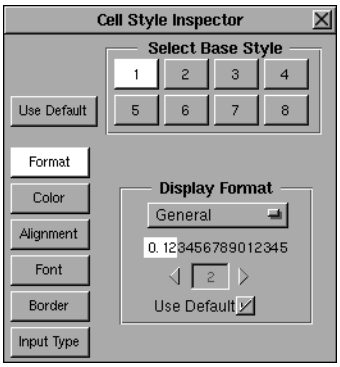
erence Panel. You may chose any width or height from 32 cells to 16,384 cells. Other options let you add or remove add-in programs for spreadsheet functions or graphs. The Preferences panel also lets you set units of measurement (inches, centimeters, points), a currency symbol, copy-data formats (EPS, ASCII, tabular), how the Return, Enter, and arrow keys will act, and whether or not macros should autoexecute.

Item	Description
Function AddIns	Allows you to select the AddIn functions that Mesa will load. An AddIn function is a small program that you can write to perform a specific type of calculation (such as bond valuing, for example), or to extend Mesa' s functionality in some other way . Note that AddIns are not available in Sunrise.
Units of Measure	The units of measure Mesa will use: Points, Centimeters, or Inches.
Macros	Enable the execution of the \0 macro or AutoExecute scripts when a sheet is loaded.
Sheet Size	The default number of rows and columns of a new spreadsheet.

Item	Description
Copy Format	When data is copied from a range of cells onto the Pasteboard, Mesa can place different information on the Pasteboard. Mesa can put an EPS image of the data on the Pasteboard so that you can paste a complete graphic representation of the Mesa range into another document. ASCII data is tab-separated values from your sheet. Tabular Data is similar to ASCII, but is a special data interchange format supported by other NEXTSTEP applications.
License	If you have enabled Mesa with a License code and you transfer the license to another person or account, remove the license enabling code from your current account by clicking on this button.
Currency Symbol	When Mesa formats a currency string, it places a character at the beginning and the end of the entry . The default is a “$” in front and nothing in back, but you can change this to suit the currency conventions in your location (for example, £).

Item	Description
Key Attributes	The pop-up lists allow you to define how the cursor will move after you press *Enter* or ↵. If "Arrow Enable" is checked, the ← and → keys will not move the cursor out of the cell while the cell is being edited.
Make Backup	Designates whether Mesa makes a backup (.Mesa~) file each time the file is saved. By default Mesa makes a backup.
Hide DB Password	Designates whether passwords will echo in the Query Inspector

Cell Style Inspector



The Cell Style Inspector, used to inspect and change the contents of a cell, reflects the style and formatting information of the current cell. Changes made via the Cell Style Inspector will affect the style of all cells in the current range. **Format → Cell Style Inspector...** (*Command-Shift-C*) opens the Cell Style Inspector.

Style Templates

Style Templates can be used to create better looking and more maintainable worksheets. Each worksheet contains eight style templates numbered 1 through 8. Each cell has a format that is derived from the style template but can be changed.

Each cell has a style template, but each parameter about the cell can be changed. By default, cells derive from Style Template 1, but this can be changed either by clicking on another style template in the Cell Style Inspector, by selecting the desired style from the **Format** → **Style Templates** menu, or by typing *Command-1* (*Command-one*) through *Command-8*. After the style template for a cell has been selected, individual formatting items in the cell can be changed from the default. If the style template itself is changed via the sheet inspector, all cells that derive from the changed style template will change all attributes that have not been over-ridden in the particular cell style.

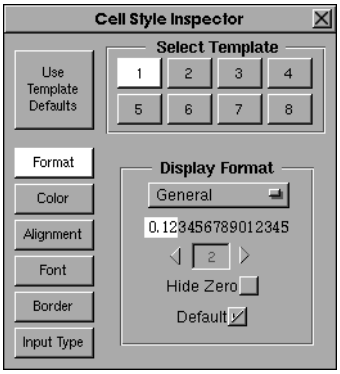
Different parts of the worksheet can be based on different style templates. Because cells default to style template 1, use that template for the basic style of the worksheet. The summary section may be in a different font (bold or larger, for example), so cells in the summary section could be style template 2. In the Sheet Inspector, change Style Template 2 so that its font is bold. Experimenting with style templates will help with these concepts.

The Inspector

The Cell Style Inspector has six views: Format, Color, Alignment, Font, Border, and Input Type. The views can be selected by clicking on the correspond-

ing button on the left side of the inspector. Each view contains a check-box designated “Use Default.” If this is selected (i.e., there is a check mark in the check box), the default format from the style template is used. If a change is made to the format, the template will be overridden and the new selection will be used for the cell. To go back to using the default item for the cell, click on the *Use Default* check box. To go back to the defaults for all views, click on the *Use Default* button in the main view or use **Format → Style Template → Use Defaults**. Changes made in the cell style inspector are applied to all cells in the currently selected range.

Format



The format view allows selection of the format used to display the value of the cell. Most items apply only to cells with a number value. Exceptions are the Text and Hidden formats. The Text format displays the input of the cell. If a cell has a formula, the formula — not the calculated value — is displayed. The Hidden format displays nothing in the cell, as if the cell is blank. This format can be used to hide intermediate results that would otherwise clutter the worksheet.

Date Int'l 1 displays date serial numbers in the format MM/DD/YY.

Date Int'l 2 displays date serial number in the format MM/DD.

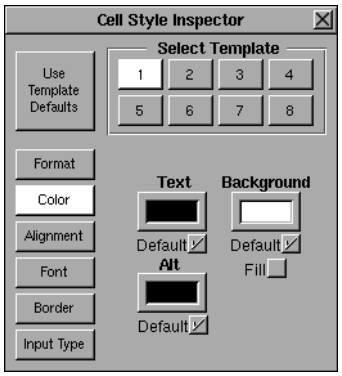
Time Int'l 1 displays the time portion of date serial number in the format HH:MM:SS.

Time Int'l 2 displays the time portion of date serial numbers in the format HH:MM.

The decimal selector changes the number of decimal places displayed when Fixed Decimal, Scientific, Currency, Comma, and Percent formats are selected.

Some commonly used formats are available in the **Format** → **Display Format** menu. See "Display Format submenu" on page 153.

Color



The information in cells is displayed in color. Normally, the contents will be displayed in the "Text Color." However, if the value is a negative number, it will be displayed in the "Alternate color." The colors may be changed by dragging and dropping a color

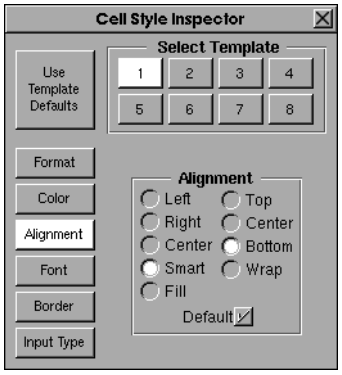
chip into the appropriate color well. The patterns may be changed by dragging and dropping a pattern chip from the pattern picker into the color well. The text color of a cell or a range of cells can be changed by dragging and dropping a color chip on the cell or range directly on the worksheet. The color picker can be opened by clicking on the border around the color well or with the **Format → Colors...** command, and the pattern picker can be opened with the **Format → Patterns...**. It may be useful to place this command on the ToolBar so that the color picker can be quickly opened.

The background or shading of a cell can be changed by *Alt-Dragging* a color chip. To *Alt-Drag* a color chip, press the *Alternate* key before you click on the color well to pick up the color chip. When you drop the color chip on a cell or range, you will not see the color change until you un-highlight it. The background pattern of a cell can be changed similarly.

By default, the cell is shaded with the background color of the sheet. To set a background fill, click on the *Fill* switch. If the background and the shading color are the same, then you will not see a change; however, if you change the background color of the sheet, the cell will retain its color. To change the cell shading, you can *Alt-Drag* color over the selected cell or range or drop the color in the *Shading* color well in the Cell Style Inspector.

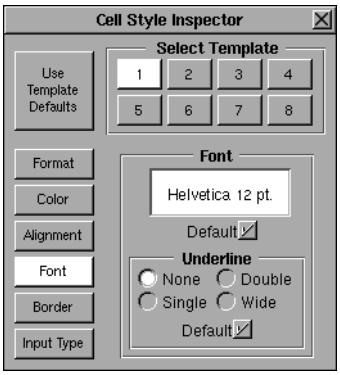
Alignment

Alignment determines where information will be displayed in the cell. The options in the left hand column refer to the horizontal alignment of the cell. Left alignment displays the cell justified with the left side of the cell. Right alignment justifies the contents to



the right. Center centers the contents. Smart alignment right-justifies numbers and left-justifies non-numeric information. Fill alignment fills the cell with repeated copies of the contents. This is useful for filling a cell with dashes or equal signs. The options in the right hand column refer to the vertical alignment of the cell. Top alignment moves the text to the topmost part of the cell, Center centers the text vertically, and Bottom aligns the text with the base of the cell. The Wrap option wraps string information in the cell down so that it does not spill out the sides of the cell, but is displayed on multiple lines within the cell.

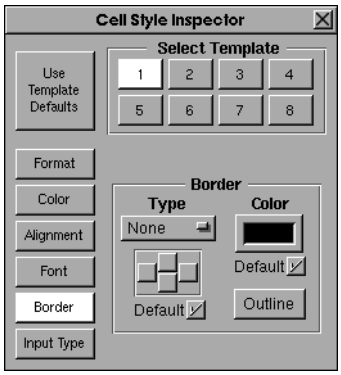
Font



This view of the Cell Style Inspector allows you to change the font by dropping a font chip in the font well. Open the Font Panel by double-clicking in the font well, by clicking the button marked *Font* , or by using the **Format** → **Fonts...** (*Command-t*) command. For more information on dragging and dropping font chips, see “Fonts” on page 157.

You can underline the contents of a cell by selecting a single, double, or wide line options.

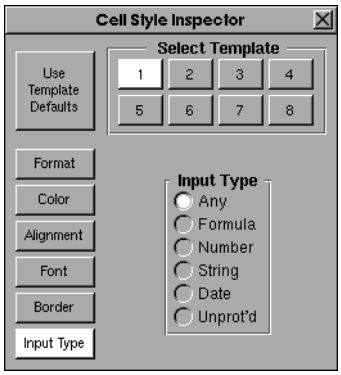
Border



Borders can be placed around cells or ranges. The border is a single or double line of a selected color that can appear on any of the four sides of a cell. The cell border is turned on and off via the diamond of check-boxes. Its width (single or double) is selected via the pop-up menu. The color can be changed by dropping a color chip into the color well. To place an outline around the currently selected range of cells, click the outline button. Single and double border outlines can also be placed around a range of cells using the **Format** → **Border** menu. (See "Border submenu" on page 153.) Single and double borders cannot be mixed within a single cell.

Note: Single and double borders appear as thin and thick borders on the computer display, but they will be printed as single and double borders on the printer.

Input Type

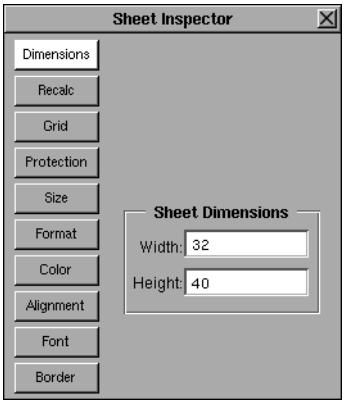


Mesa allows basic checking of data input. Cells can be designated as Numeric, String, Date, or Formula and that cell will accept only input of the designated type. If a cell is typed *Any* , then any type of data is allowed for input. *Unprotected* cells are like cells typed any, yet data can be entered into an unprotected cell even when sheet protection is enabled.

Sheet Inspector

The Sheet Inspector changes items that affect the whole worksheet or all cells in the worksheet. For example, the Sheet Inspector is used to change the format of a Style Template that cells derive from. (See “Style Templates” on page 76). The **Format** → **Sheet Inspector...** command opens the Sheet Inspector. The sheet inspector displays and changes parameters of the currently active worksheet. When all worksheets close, the Sheet Inspector will automatically close because it has nothing to inspect.

Dimensions

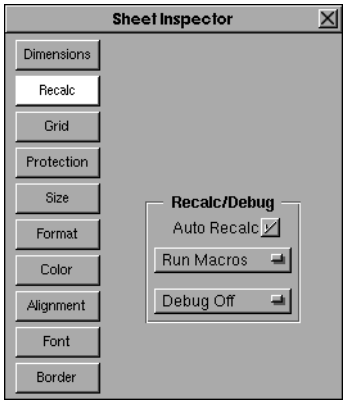


Mesa worksheets can have up to 16,384 rows and columns. However, when a worksheet has this many cells, using the scrollers to move around the worksheet becomes very clumsy. The number of rows and columns displayed in the worksheet can be viewed and changed in the Dimensions view of the Sheet Inspector. To change the dimensions, enter the desired number of rows or columns in the height or width boxes. Mesa worksheets also grow automatically when you move beyond the edge with the arrow keys. Demo versions of Mesa do not allow the dimensions to be changed.

Note: The dimensions of the worksheet affect only how the information on the worksheet is displayed. A worksheet may display only 32 rows and columns, but may contain data in any cell in the worksheet. The data in cells that is not displayed is not lost and can be used in formulas and viewed by setting the worksheet

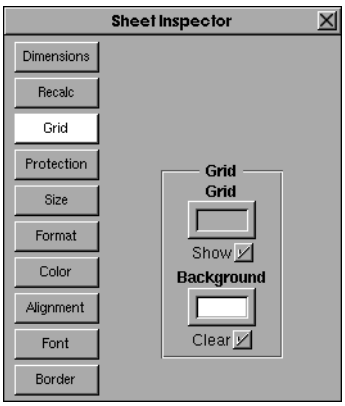
dimensions to the appropriate size. There is no speed or storage size penalty associated with larger dimensions.

Recalc



Mesa always uses “natural order” recalculation so that the one answer for a given cell is returned correctly even if cell data in other parts of the worksheet are accessed. However, Mesa cannot resolve circular references in cell formulas. Circular Reference debugging is provided to track and allow you to fix circular references. To enable calculation debugging, select “Stop on Error” or “Stop on Circ.” from the pop-up. To step or trace through macros, select the appropriate item from the pop-up. By default, Mesa recalculates the entire worksheet each time data in the worksheet is changed. For large worksheets, this could take a second or two. To disable automatic recalculation, uncheck the “Auto Recalc” check-box. If Auto Recalc is disabled, remember to use the **Sheet → Recalculate** (*Command*-=) command to update the worksheet after data has changed.

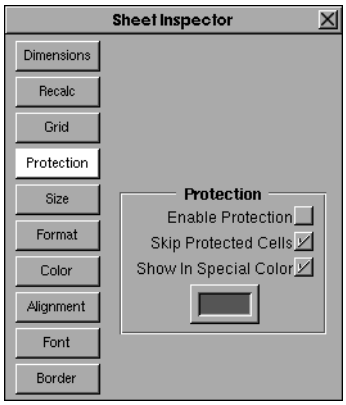
Grid



The “Show” check box turns the worksheet’s grid on and off. The color of the grid can be changed by dropping a color chip in the color well. You can change the background color of the entire worksheet by dropping a color chip inside the *Background* color well. All cells that do not have background shading will display this color as their background. If the “Clear” check box is checked, the background will appear white on the screen, but if the cells are copied and pasted to another application, the background from that application will show through. This is useful for creating presentations with Concurrence™ or Create™.

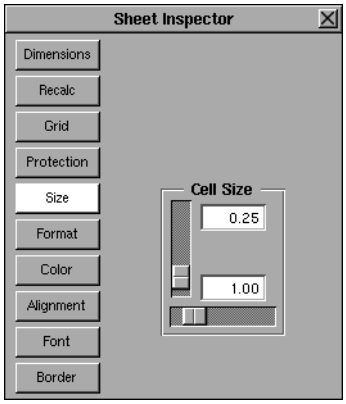
Protection

Cell Protection is used to block data entry into cells that are not specifically typed Unprotected. (See “Input Type” on page 83). To enable cell protection, check the “Enable Protection” check box. If protection is enabled, protected cells will be skipped when tabbing through a range if “Skip Protected Cells” is selected. Protected cells will be displayed in a special



color if “Show in Special Color” is selected and the worksheet is protected. The color of the protected cells can be changed by dropping a new color into the color well.

Size



This view allows the default cell height and width to be changed. The current default width and height is displayed in the current measurement units. To change the width or height, use the sliders to select the desired default width or height, or type the desired size directly into the height and width boxes.

Changing the default will not change the size of rows or columns that have been resized. For more information, see “Row and Column Size submenu” on page 154.

Format, Color, Alignment, Font, Border

These views change the style templates that cell styles derive from. See the above descriptions of formatting for specific information about the effects and features of these views.

Graph Inspector

The Graph Inspector allows you to inspect and change attributes of charts. See “The Graphics Menu” on page 167 for more information on how to create charts. To inspect a chart, select it by clicking on it. Choose **Graphics → Graph Inspector...** to open the Graph Inspector. You can also double-click on a chart to open the inspector. The Graph Inspector disappears when a graph is no longer selected and reappears when the graph is selected again.

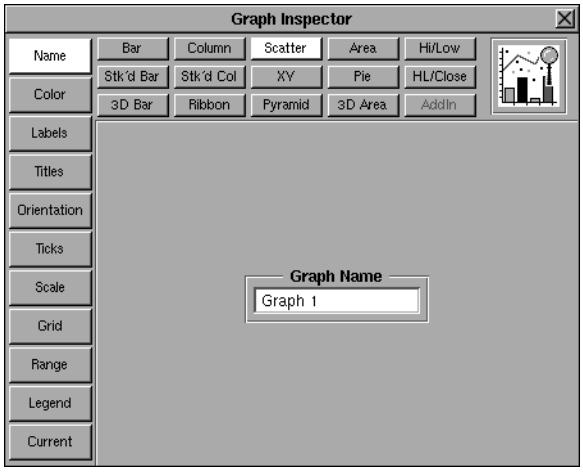
The Graph Inspector has fifteen buttons at the top of the inspector; one for each of Mesa’s graph formats, plus an *AddIn* button. The button corresponding to the current graph type is highlighted. To change the graph from one format to another, simply click the button corresponding to the desired format.

There are a few differences between the way flat graphs and 3D graphs work. 3D graphs can be rotated using the *Current* view. While ordinary graphs can only have one set of labels, 3D graphs can have two sets of labels, but the labels on 3D graphs do not rotate.

The Graph Inspector allows you to invoke special features in Mesa's graphs, including graphing discontiguos ranges, adding labels, and creating mixed bar and line charts. The different views of the Graph Inspector allow you to adapt different parts of your graph to suit your needs.

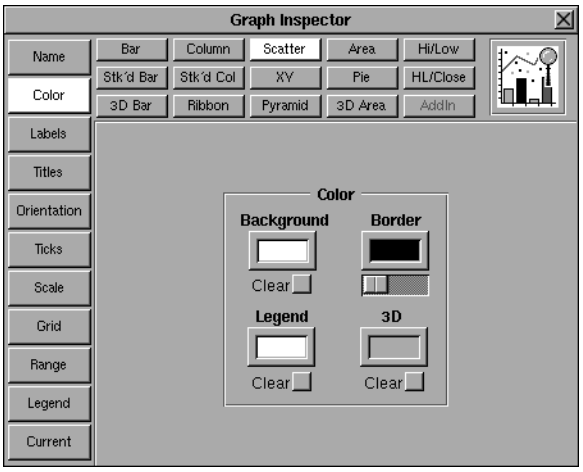
The Graph Inspector contains nine basic views plus one view that is specific to the type of chart. This last view is called *Current* because it changes depending on the current chart type.

Name



Each chart has a unique name. You select a chart by name using **Sheet** → **Goto Cell** (*Command-g*). You can also print a chart by name in a report. To change the name of a chart, select the name, be sure that the Graph Inspector is selected (i.e., the title bar is black not light gray), type the new name, and press ↵. You cannot give a chart the same name as another chart in the worksheet. If you try to, Mesa will beep and the name will not change.

Color

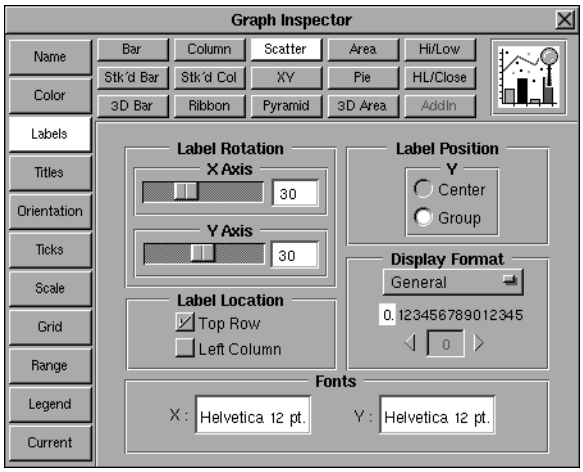


The *Color* view allows the background color, border color, legend background color, and 3D background color to be changed. If the "Clear" check box is checked, the background below the graph will show through. This is especially useful for developing presentations using Concurrence™ or Create™. The width of the border can be changed using the slider below the border color well.

To change the color or pattern of the plotted part of the graph you do not need to use the Graph Inspector. Simply drag and drop color chips onto the graph itself. Mesa will keep color shading and patterns consistent for data that is grouped together.

Labels

The different areas of the on the *Label* view if the inspector allow you to change the appearance of the graph's labels. Labels are placed near the horizontal and vertical axis of a graph.



You can adjust the rotation of the labels using the *Label Rotation* area. Rotating labels enhances the visual appearance of the chart and can be used to display labels that would otherwise overlap. Labels may be rotated using the sliders or by typing the desired rotation in the text fields and pressing ↵.

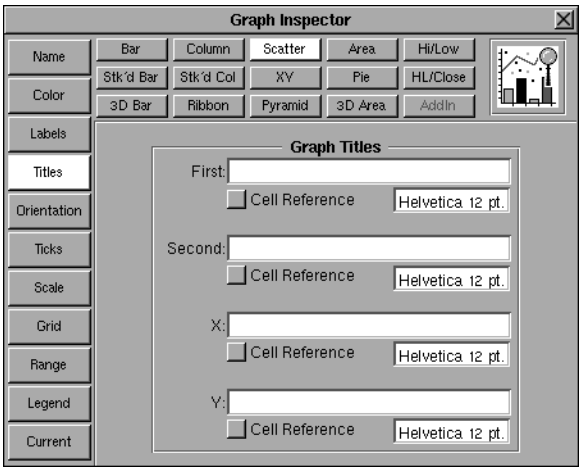
The check boxes in the *Label Location* area allow you to designate whether the top row and/or the left column contains additional labels. You can also designate ranges containing additional labels using the *Range* view of the Graph Inspector. Flat graphs can display one set of additional labels, and 3D graphs can display two sets of additional labels.

The font wells in the *Fonts* area of the view display the current font used to draw the labels on the X and Y axis. To change the font, open the Font Panel by selection **Format → Fonts...** (*Command-t*) or by double-clicking in the font well. Build the desired font and drop the font in the appropriate font well. The fonts of the X and Y axis can also be changed by dropping a font chip on the X or Y axis or any label of the chart.

The *Display Format* area of the view allows you to change the format and precision of the axis labels. You can select any of the Mesa's standard format, including date formats. This will not affect the format of additional labels.

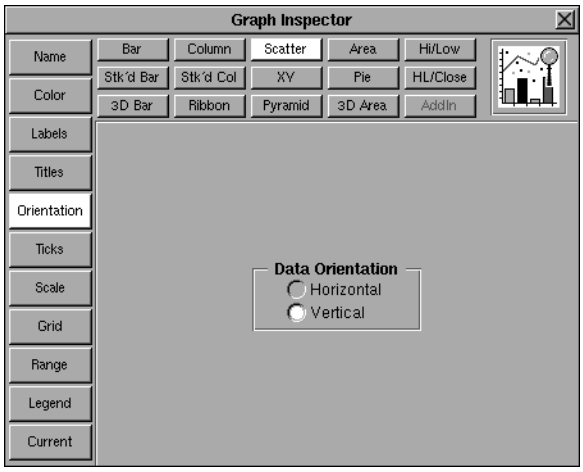
The *Label Position* area allows you to select the vertical centering of labels drawn on the Y axis. Select the setting that you prefer.

Titles



Charts can display titles on the top, bottom and left side. The *First* and *Second* titles are drawn at the top of the chart. The *X Axis* title is drawn under the X axis. The *Y Axis* title is drawn to the left of the Y axis and is rotated 90 degrees. Enter the text to appear in the titles. The font of each title can be changed by dropping a font chip in the font well or by dropping a font chip on the title of the chart. If *Cell Reference* is checked, the text is interpreted as an address and the contents of the cell at that address is displayed.

Orientation



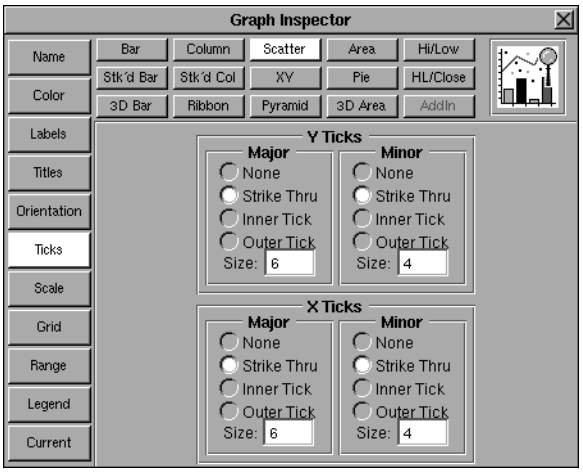
The *Orientation* defines the order that the data will be accessed when drawing the chart. Horizontal orientation reads data "across" and additional data ranges are added under the base range. Data in a column will show up with the same color and pattern. Vertical orientation reads data "down" and additional ranges are added on the right side. Data in the same row is given the same color and pattern. Mesa tries to choose the correct orientation for your data when you create a graph.

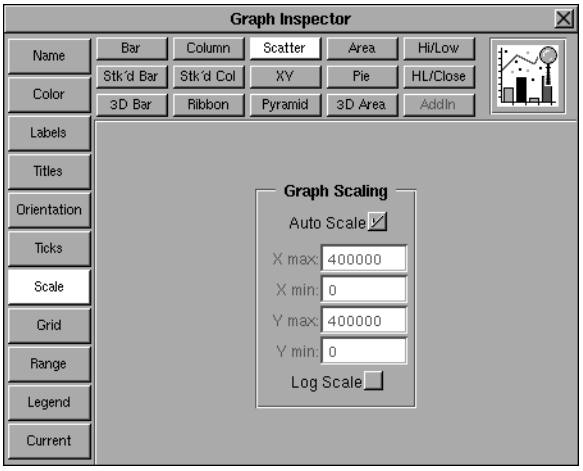
Ticks

This view allows you to set how the format in which ticks on the X and Y axis will be drawn .

Scale

By default, Mesa automatically chooses the largest and smallest X and Y values for data plots. You can change these values manually to enhance the

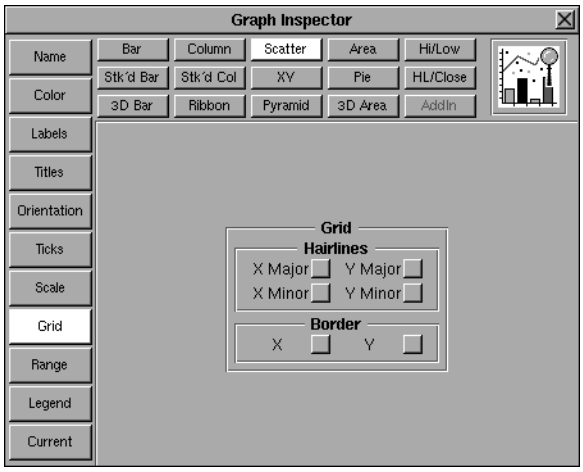




appearance of a chart. To turn off automatic scaling, click the *Auto Scale* check box. Enter the desired scale values in the text fields and press ↵. The minimum and maximum values must be different.

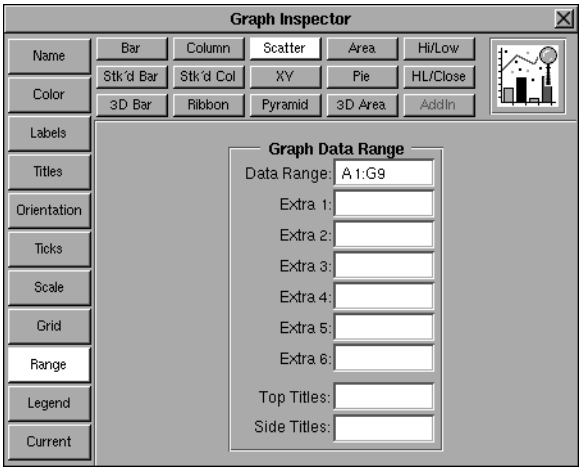
Graphs are displayed in linear scale by default, but can be displayed in log scale by clicking the “Log Scale” check box.

Grid



This view allows you to control the display of a light gray grid in the background of the chart. You can also select an optional black border around the graph area on the chart.

Range

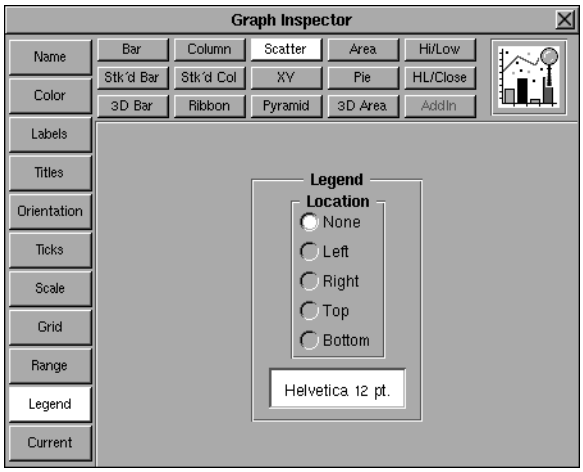


You can add up to six discontiguos ranges to the base range of a data plot. Additional ranges will be added at the bottom or right side of the prior data range depending on the orientation of the chart. (See "Orientation" on page 93.) To add ranges, type the range into each of the six provided text fields.

In addition you can also add ranges containing titles to your graph using the *Top Titles* and *Side Titles* fields. The contents of these ranges will be added as additional labels to the graph the same way that selecting *Top Row* or *Left Column* in the *Label Location* area of the Label view would, except that the ranges entered as titles in this range do not need to be in a specific location in relation to the rest of the data.

Legend

This view allows you to place a data legend on the chart. The legend can be located on top, below, or on either side of the chart. The legend's font can be



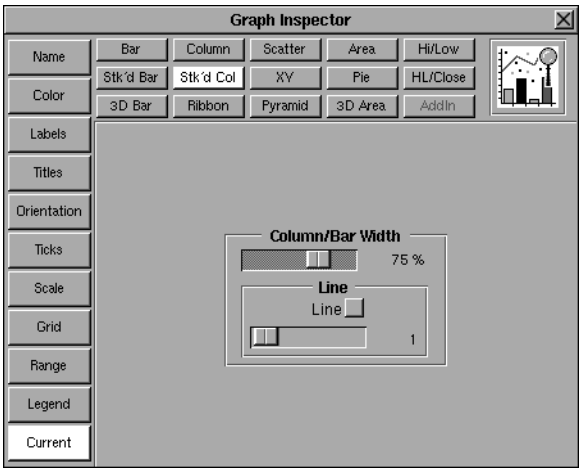
changed by dropping a font chip in the font well or by dropping a font chip directly on the legend on the chart. The color of the chart can be changed by dropping color chips on the legend or by dropping the color chips on the item that you want to change directly on the graph.

Current

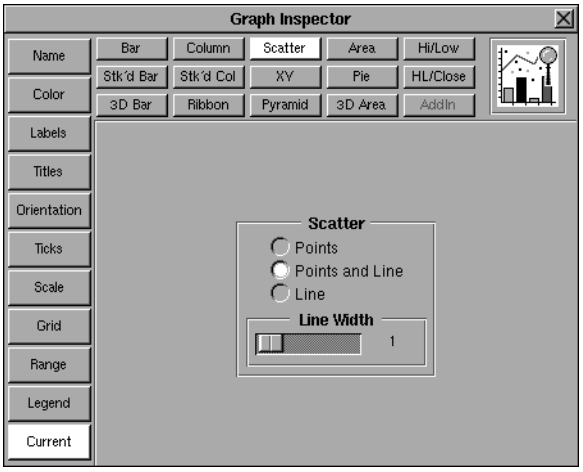
Some chart types have special inspector views to change unique parameters. These views appear in the *Current* view depending on the chart type.

Bar and Column Charts

This view allows you to determine the width of the bar or column by moving the slider to select the desired width. The line option will plot a line (the first series of data) on the same graph. If the orientation of the graph is horizontal, the line is plotted from the data in the left column. If the orientation of the graph is vertical, the line is plotted from the data in the top row.

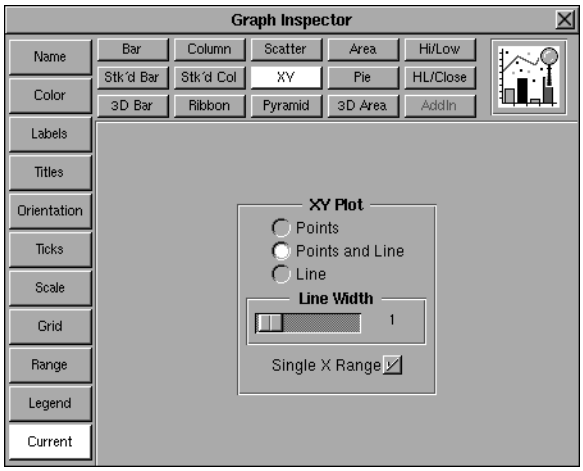


Scatter Charts



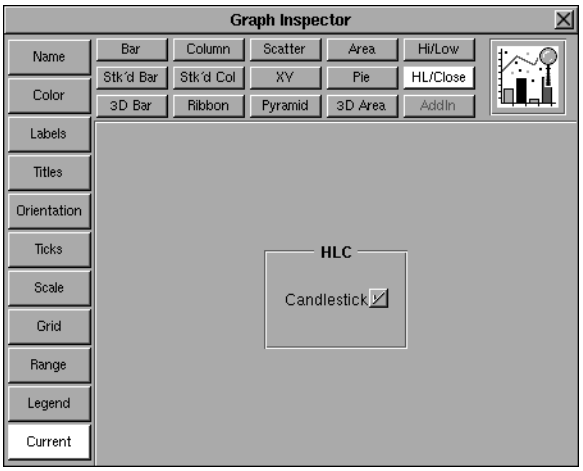
Scatter charts can be drawn with data points, lines, or both. Select the drawing method using the radio buttons. Use the slider to adjust the line width.

XY Charts



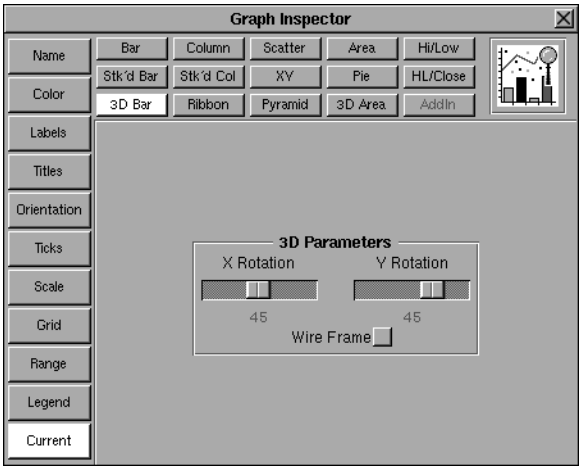
In addition to the chart options for lines and data points that Scatter Charts allow, XY charts allow you to select whether the Y values of the plot will be matched with a single set of X values or if they will be matched with the X values of the adjoining row or column. The default is a single X axis.

High/Low/Close Charts



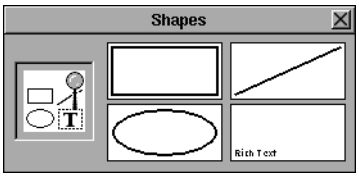
High/Low/Close charts have an option for displaying candlestick graphs. This option is chosen by default.

3D Charts



3D charts have sliders for adjusting the X and Y rotation of the graph and a wire frame option.

Shape Panel

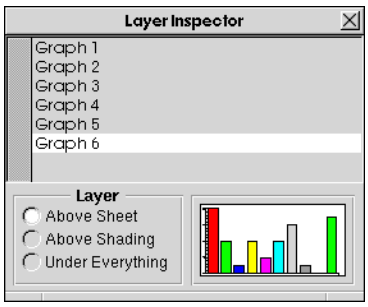


The Mesa Shape Panel displays several different shapes. The shapes can be dragged onto the worksheet. To move a shape, select it by clicking on the shape outline, and drag it to the desired location. The dimensions of the shape can be changed by selecting the shape and dragging the handles. The text shape will be selected the first time you click on it. To edit it, click on it a second time.

If you double-click on the a shape, the shape inspector will open. The Shape Inspector allows you to change the color and width of the border, and the color of the shape's interior.

Layer Inspector

The Layer Inspector gives you more options for displaying graphics. Graphs and other images can be placed above everything else in the worksheet, under everything, or above the background but below the

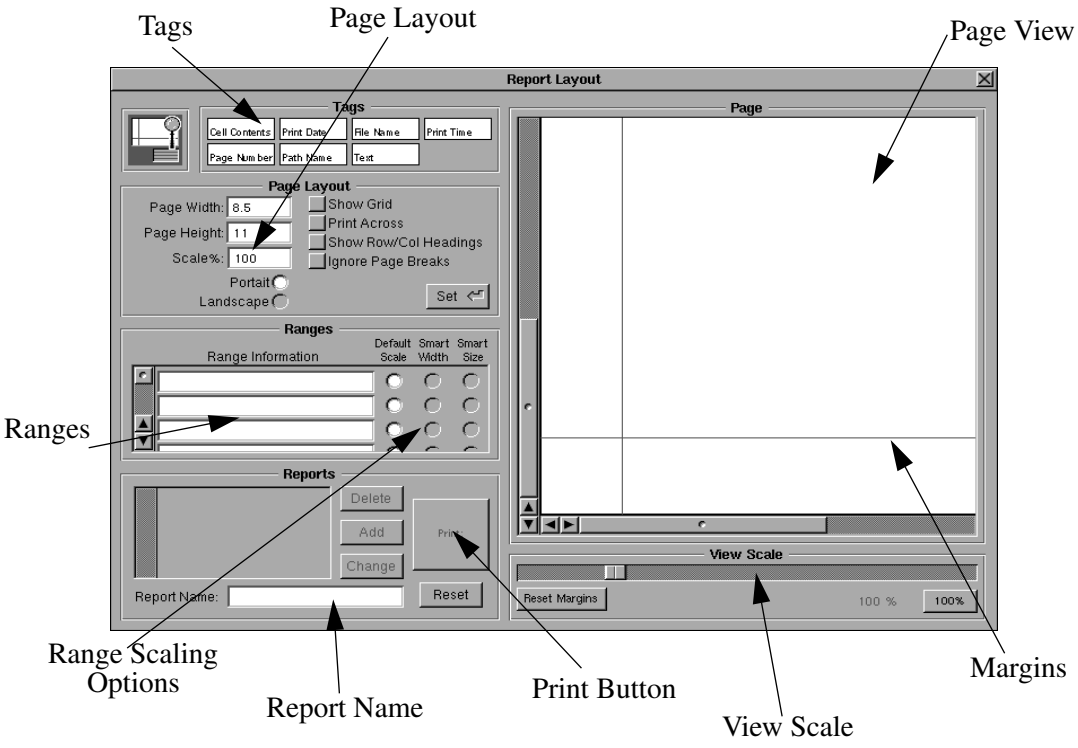


values in the cells. TIFF and EPS images can also be placed below the sheet. This is useful for placing corporate logos and other images on worksheets for developing presentations.

Report Layout Inspector

Mesa allows you to build reports using the Report Layout Inspector. A report is named, then the layout of the page and its contents are defined. Finally, the

data ranges and graphs are selected. Use the **Format → Report Layout...** (*Command-Shift-R*) command to open the Report Layout Inspector.



The following table briefly summarizes the elements of the Report Format panel.

Item	Description
Report Name	Each Report has a name. That is how you tell Mesa which report you want to print or change. Report names should describe the report, e.g., Q1_Sales, Checking, Summary_Report.

Item	Description
Print Button	After a report has been added, you can click on the print button to print the currently selected report. The Print Button is a tool that you can *Command-drag* to the T ool-Bar or a T ool Holder in a window .
Ranges	A range is a group of cells or a graph that you want to print. Mesa starts printing each range on a new page. A report can contain up to 32 ranges
Range Scaling Options	The range scaling options *Smart W idth* and *Smart Scale* change the size of the font used in printing the ranges. *Smart Width* will adjust the printed range so that it is only one page wide. *Smart Scale* will adjust the scale so that the entire range prints out on a single page.
Page V iew	A report starts as a blank page. Ranges are printed within the page margins. In addition, you can place tags on the page. Y ou should place tags outside the margins so that they are not printed over . Tags contain information that you want to appear on every page such as print date or time, file name, or some text that you enter .
View Scale	A full-sized page is too big to be displayed in the Report Format panel. Y ou can change the scale of the page (miniaturize it) so that you can see more of the page, or “zoom in” and work on a specific part of the page.

Item	Description
Margins	The margins are the borders on the page within which ranges are printed. You can adjust the margins by moving the mouse over a margin, clicking and dragging until the margin is sized to your satisfaction. If you hold the *Command* key while you are dragging the margin, Mesa will adjust the corresponding margin to the same size.
Page Layout	The Page Layout allows you to change page attributes. For example, you could print in landscape mode on legal paper by entering 14 for the paper width and 8.5 for the height and clicking "Set."
Tags	Tags can be dragged and dropped onto your page and moved around. Tags contain information such as print time and date, file name, page number, etc. Tags are printed on every page. Be sure to place tags outside the margins so that they are not covered by the ranges.

Report names

When you custom design your printed output, you create a *Report* that you name. The Report is saved with the worksheet file and can be used over and over again. You can create as many custom reports as you need. When you are working on a particular worksheet, the names of all of its reports appear in the *Reports* box on the Report Layout panel whenever you display it.

Creating a new report

Start out by giving your new report a name. To assign a name, click in the *Report Name* text field on the panel and type a name. You can call it *Report1*, or *Accounting_print_report*, or anything you want.

To record the report name, click the *Add* button and the name will appear in the list of defined report names.

Ranges

The list of ranges is located above the Reports area on the Report Layout panel. You can enter up to 32 print ranges. A range can be a group of cells or a graph. If a range is a group of cells, it can have titles on the top and on the left side. A range can also be a graph. Each range is printed on a new page. Cell ranges can spill over to other pages. To print a frozen set of titles on the top or side of a range, follow the range with "T=" or "S=" and the range of the top or side titles. The "T" and "S" are case insensitive.

To enter a range, simply click on the first empty line in the Ranges box to position the text cursor. Then type a range and press ↵ to record it. Notice that the text cursor moves to the next empty line so that you can enter additional ranges. The following table gives some examples of ranges.

To Print	Enter
Cells in the range A1:D10	A1:D10
Cells in range A2:D50 with row 1 printed at the top of each page	A2:D50, t=A1

To Print	Enter
Cells in range B2:G72 with row 1 and column A printed at the top and on the side of every page	B2:G72, t=A1, s=A1
Cells in E2:G72 with rows 1 and 2 printed at the top of every page	E2:G72, T=A1:A2
Print the range named Sales	Sales
Print the range named Sales with row 20 printed at the top of every page	Sales, T=A20
Print the graph named Graph 2	g=Graph 2
Print the range named Actual with columns D and E printed to the left of the range on every page.	Actual, s=D1:E1

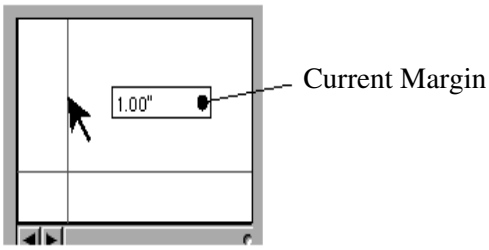
To locate a range that you wish to edit, move the highlight up the list with *Shift-Tab*, or down the list with *Tab* or ↵. (When the highlight reaches the bottom of the list, it will wrap back to the top again.) Each range is printed on a separate page. Large graphs are scaled to fit on the page.

Setting the margins

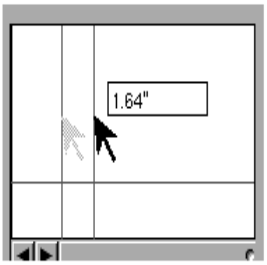
Margins define the area in which ranges are printed. You change margins by dragging the margin to the desired setting. You can change corresponding (Left and Right or Top and Bottom) margins at the same time by holding the *Command* key while setting the margin. While you drag the margin, Mesa displays a

window containing the size in current units (set via the **Info** → **Preferences** command) of the margin. To change the margin of the current report:

- Use the left and bottom scroll bars to scroll the page so that you can see the margin that you want to change.
- Point to the margin, press the mouse button, and observe the box showing the margin' s size.



- Drag the margin to its new position and release the mouse button.



Note: You can set the left and right (or top and bottom) margins at the same time by pressing the *Command* key while changing one margin.

To restore the margins to their default values, click the *Reset Margins* button.

Changing the page size

The default page size displayed in the *Page Setup* box on the Report Format panel is 8.5” by 11” (letter size). To change the page size to 8.5” by 14” (legal size) you can change the page height as follows:

- Double-click the number 11 in the *Page Height* box to highlight it.
- Type in `14` and press ↵ or click the *Set* button.

You can adjust the page width in the same way.

Mesa lets you define custom pages as small as 2.5” by 2.5” and as large as 100” by 100”. Your printer may not be able to handle pages as large as you define them. If you are using the NEXT printer, do not create pages larger than 8.5” by 14”.

Adding titles

Row borders

Suppose you want to print a table with rows 1 and 2 appearing at the top of each page.

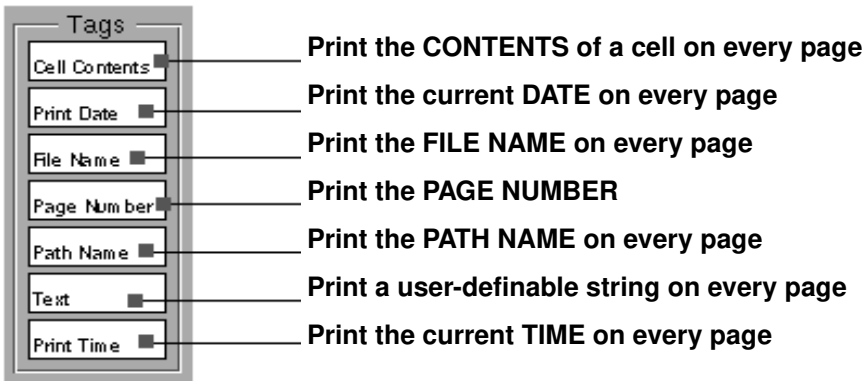
With Mesa, you can specify border ranges when you enter your print ranges. A top border range of A1:E2 is specified by typing a space and *t=a1:e2* following the print range in the *Ranges* box on the Report Format panel. The complete range entry would be: *a3:e80 t=a1:e2* .

Column borders

Column or side borders work the same way as row borders. A side border range of A1:B10 is specified by typing a space and *s=a1:b10* following the print range in the *Ranges* box on the Report Format panel. The complete range entry would be: *c1:f10 s=a1:b10* .

Adding headers and footers

Mesa uses report *Tags* to let you put special headers anywhere on a page. Tags print the specified information on each page. They also allow you to place items of information (text, page numbers, dates, etc.) anywhere on the page. You can select from the following tags:



For example, to add a Cell Contents tag to your page layout:

- First, use the *View Scale* slider below the Page box to reduce the page image so that you can see the whole page in the box (about 38% should do it).
- Drag a Cell Contents tag from the *Tags* list to the top of the page and position it above the margin line. Release the mouse button when the tag is where you want it.

- Restore the page image to normal size by clicking the *100%* box below the scale slider .
- Scroll the page view to display the Cell Contents tag. It contains the default address *A1*.
- Double-click the tag to highlight the address, enter a new cell address, and press ↵.

Add other tags in the same way. You can change the font and color of tags by dragging and dropping font and color chips on top of the tags that are on your page. When you change the font of a tag, please be sure to size the tag so that its text is not clipped by the border of the tag.

Creating and printing custom tags

Mesa allows you to create custom tags for instances where you might want to combine text with other tags, like dates or page numbers.

For example, to create a tag that prints "This is page 8 of 12," you would:

- Drag a *Text* tag from the T ags list and position it on the page image.
- Click on the tag border to display its "handles" and enlarge it enough to display your text.
- Enter your desired text and any of the special format symbols (listed in the table below) and press ↵.
- Resize the tag, if necessary , using the tag handles as before.

- The tag contents for the above tag would be: `This is page &p of &n.`

Mesa T ag Formatting Codes

Format	Mesa Code
Page number	&p
Total # of pages	&n
Current date	&d
Current time	&t
Document name	&f
"&"	&&
Cell contents	Use the "Cell Contents" tag

Using special fonts and formatting

You might want to use a different font in a report tag for stylistic reasons.

Suppose you have created the page number tag described above. To change the font:

- Select **Format** → **Fonts...** (*Command-t)* to display the Font Panel and move it to a convenient place on the screen.
- Select a new typeface and size and click the *Preview* box on the Font Panel.
- Repeat this process to choose the font that you want.
- Move the mouse pointer into the font preview box and press the mouse button. You will see a small square with the letter *f* in it. This is a *font chip* .

- Drag the font chip to the tag box and drop it in. The font of the text will change.
- You may need to resize the tag box (with its handles) to be sure that all of the text is visible.

You can change the color of a tag just like changing the font. Simply open the color panel with **Format → Colors...** (*Command-t*), choose the color, and drag & drop the color chip onto the tag. The background color of the tag can be changed by *Alt-dragging* a color chip onto the tag.

Printing landscape or portrait

To print in a landscape orientation, simply click the *Landscape orientation* button in the Page Setup box on the Report Format panel.

To return to portrait orientation, click the *Portrait orientation* button.

Remember to click the *Set* button after changing the orientation.

Other printing options

- Show Grid: This option will print the worksheet grid for each range.
- Print Across: This option will cause multiple-page ranges to print across before printing down.
- Show Row/Col Headings: This option will show the worksheet row and column headings for each range.
- Ignore Page Breaks: This option will cause any page breaks set within the worksheet to be ignored. Page breaks will still occur between ranges. For in-

formation about setting page breaks in the worksheet, see "Page Breaks submenu" on page 156.

Printing and choosing a printer

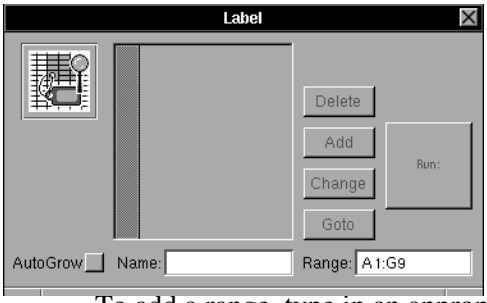
Once you have specified a range and a printer to print to, you can print your work.

- Display the Print dialog panel (*Command-p*).
- If you want to print to a specific printer , select that printer in the Print Dialog.
- If you want to print a specific report, select the report name from the list in the "Choose Report Format" list.
- Click the *Print* button to print the report, or you can choose to fax, preview , or save the file as a PostScript image.

The print button on the Report Inspector will print the highlighted report. The first time you use the print button it will open up the Print dialog panel. After that it will automatically use the last device that the report was sent to. If you would like to send a report to a different device, select **Print...** form the main menu. This will open the NEXTSTEP Print Panel. Select the report name from the Choose Report Format box on the Print Panel.

Label Inspector

The Label Inspector allows you to view and change labels. Labels are names that are assigned to a rectangular range of cells. Ranges can be designated as *AutoGrow* . The **Sheet → Labels → Label Inspector...** (*Command-Shift-L*) command opens the Label Inspector and sets the range field to the current cell or current range.

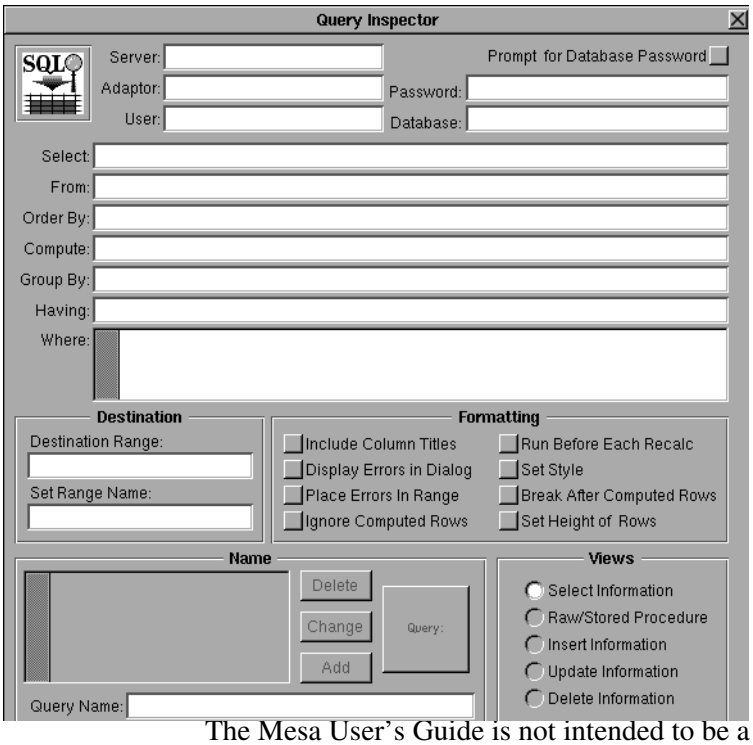


To add a range, type in an appropriate name, press *Tab* , enter the cell range for the label, and press ↵. The range will be added to the list. To change a range, click on the range in the list. The name and cell range will be placed in the text fields. Make the appropriate changes and click on *Update* . To remove a range from the list, select the range and click *Delete* . To select a range on the worksheet, select the range name, then click *Goto* .

Labels are also used to keep track of the location of macros. To execute a macro, select the label from the list and click on the *Run:* button. The *Run* button may be *Command-dragged* to either the ToolBar or to a tool holder in a window.

Query Inspector

The Query Inspector allows you to build SQL queries that are sent to an SQL database server. The data returned from the database is placed in a range on the worksheet. Using the query inspector, you can retrieve data from a database and use Mesa to build models based on that data and as a report and presentation tool.



The Mesa User's Guide is not intended to be a guide to SQL. This section of the manual will explain what the different fields are. For more information on building queries, please refer to a guide to an SQL manual.

The Query Inspector has a series of views that allow

Mesa Query Inspector

Item	Description
Server	The server field designates the name of the database server to log onto and send queries to.
Adaptor	The adaptor field designates the database adapter to be used. If this field is left blank, or the name "SYBASE" is used, Mesa will use its built-in Sybase Adapter . Otherwise, and AdaptorAddIn will be used.
Password	The password that will allow the user access to the server . The password will echo unless you change the default in the Preferences panel.
Prompt	The password is stored un-encoded with the worksheet. If the server password is to remain secure, it should not be kept with the worksheet. If *Prompt* is checked off, the user will be prompted for the server password the first time the server is accessed.
User	The name of the user . If this is left blank, the user 's UNIX name will be used.
Database	If the database that the information is being extracted from is different from the user 's default database, enter the new database name here. If a database is selected, Mesa issues a "use" command before it does the query .

Mesa Query Inspector

Item	Description
Name	Each query has a unique name. An unlimited number of queries can be associated with each worksheet.
Query: Button	Clicking on this button executes the current query . This button may be *Command-Dragged* into the T ool-Bar , a T ool Holder , or onto the worksheet itself. Placing a query in a Tool Holder or on the worksheet makes it convenient to execute.

you to build queries that will retrieve data from the database via a select statement, send a raw SQL string to the server and populated the spreadsheet with the result (this is useful for executing *Stored Procedures*), add information to a table via the Insert SQL command, update rows in a table via the Update SQL command, or remove rows from the table with the Delete SQL command. You can designate the type of query to execute by selection the appropriate type of query from the list in the lower right corner of the query inspector.

This table explains the various fields for the select statement.

Mesa Query Inspector - Select

Item	Description
Select	The SQL “Select” field. This designates the columns to be selected. If nothing is entered, Mesa selects “*” (all columns).

Mesa Query Inspector - Select

Item	Description
From	The table that the columns are to be selected from. If the *From* field is left blank, Mesa selects from the "sysobjects" table.
Order By	The SQL "Order By" field.
Compute	Enter any SQL compute fields here. If you want the compute fields to be placed in the destination range, make sure that *Ignore Computed Rows* is not checked.
Group By	The SQL "Group By" field.
Having	The SQL "Having" field.
Where	The SQL "Where" field.

These options are available for both Select and Raw

Mesa Query Inspector - Select and Raw Options

Item	Description
Destination Range	The range of cells or named range in which to place the data resulting from the query .
Set Range Name	This named range label will be set to the range which has received the query information. This is useful for basing reports or calculations on the results of a query .

Mesa Query Inspector - Select and Raw Options

Item	Description
Include Column Titles	Mesa places the column titles for the selected columns in the first row of the destination range if this item is checked.
Display Errors in Dialogs	Places error messages in a dialog panel.
Place Errors in Range	Places error messages in the current row of the destination range.
Ignore Computed Rows	Only place non-computed rows in the destination range.
Run Before Each Recalc	If this option is selected, Mesa runs the query before each recalculation of the worksheet. This is handy if the data in your database changes frequently or if the query is based on the value of cells in the worksheet. However , running the query may be time consuming. Mesa will run this query before running each data point in a Scenario.
Set Style	This option will set the formatting of normal rows to Style T emplate 7 and the formatting of calculated rows to Style T emplate 8. For more information, see “Style T emplates” on page 76
Page Break After Computed Rows	This option sets a horizontal page break after computed rows in your query . For more information about page breaks, see “Page Breaks submenu” on page 156

Mesa Query Inspector - Select and Raw Options

Item	Description
Set Height of Rows	This option will set the rows of the destination range to the best row size after the query . This is useful for the formatting of reports — columns can be set to "W rap" alignment and the text will flow well after a query . For more information on these formatting options, see "Row and Column Size submenu" on page 154 and "Alignment" on page 79

queries. Both of these types of queries can potentially return values that are placed in the destination range on the worksheet.

The only unique field for Raw SQL is the SQL statement itself which can contain any valid string. This statement will be sent to the SQL server to be executed. If any values are returned, they are placed in the destination range.

There are two options for the Insert, Update, and

Mesa Query Inspector - Insert, Update, and Delete

Item	Description
Display Errors in Dialogs	Places error messages in a dialog panel.
Run Before Each Recalc	If this option is selected, Mesa runs the query before each recalculation of the worksheet. This is handy if the data in your database changes frequently or if the query is based on the value of cells in the worksheet. However , running the query may be time consuming. Mesa will run this query before running each data point in a Scenario.

Delete views. These allow you to control error handling and when the query is executed.

These fields are available for the Insert view.

Mesa Query Inspector - Insert

Item	Description
Insert into	Designates the table that the values will be inserted into.
Values	The values to be inserted into the table.
Subquery	An optional subquery .

The update statement has four fields that will be

Mesa Query Inspector - Update

Item	Description
Update	The table that the update statement will act on.
Set	The columns and values to set.
From	An optional from field.
Where	The where clause that limits the update statement so it does not operate on all the records in the table.

sent as part of an update query.

The delete statement has three fields.

Mesa Query Inspector - Delete

Item	Description
Delete	The delete clause.
From	The table to remove rows from.
Where	The where clause that limits the delete statement so it does not remove all the records in the table.

The information in any other fields (with the exception of the Destination Range and Set Range Name) can be text that is typed in, the contents of a cell, or a combination of both. To place the contents of a cell into a query field, place *European* quotes around the cell or range name to be placed in the query. To access the *European* quotes, type *Alternate-Shift-9*

and *Alternate-Shift-0* . If A1 = 5 and you place `qty > «A1»` in the *Where* , Mesa will send `qty > 5` as the *Where* clause to the SQL server. Mesa does not supply quotes around text fields.

Note: The *Add* button will remain disabled until you have entered a name for the query . The *Query:* button will remain disabled until a pre-added query is selected.

Note: The Query Inspector may be disabled for a user by writing a value of YES to NoQueryInsp in the user 's defaults database.

Scenario Inspector

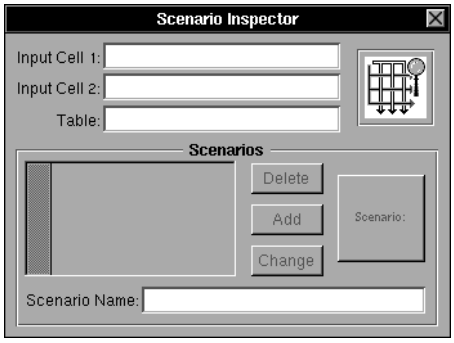
Mesa scenarios allow you to use your spreadsheet to make calculations based on more than a single formula and set of variables. For example, you might want to calculate the monthly payments and total payments on a loan for payment plans with different interest rates. You could also use scenarios to find the orbital angular momentum of a body given different theoretical values of its semi major axis and orbital period. Scenarios are especially useful when you are concerned with a set of predicted or theoretical values and varying conditions.

Mesa has two types of scenarios. Single Input scenarios show the result of changing the value of a single variable in different formulas. Double Input scenar-

ios compute the result of a single two-variable formula for different values of each variable. Both types of scenarios are created using the Scenario Inspector.

The Scenario Inspector creates a scenario table. A single input scenario will create a table which shows the result of using each different value of the variable in each formula. Double input scenario tables show the result of each different set of variables in the equation.

Using the Scenario Inspector



Mesa scenarios are calculated using the Scenario Inspector. To open the Scenario Inspector select **Sheet** → **Database** → **Scenario Inspector...** in the menu. Give the Scenario a name by typing it into the space at the bottom. Enter the cell name for the input cell or cells, and the range of the scenario table. Once the information is entered, click on the *Add* button. The name of the Scenario will appear in the list in the middle of the Scenario Inspector. You can

remove, change, or run a Scenario by clicking on its name in the list, which will highlight it, and then clicking the appropriate button.

Note: Mesa runs all "Run before each recalc" Sybase queries before running each data point in a scenario. Scenarios recalculate the worksheet after they execute, even if auto-recalc is off.

Creating Single Input Scenarios

To create a single input scenario, choose a column to enter the different values of the variable into. Leave the first cell of the column blank and enter the values starting with the second cell of the column. For example, if you wished to make a scenario using ten different values of a variable starting in the upper left corner of the spreadsheet you would leave cell A1 blank and enter the values in cells A2 through A11. Next, enter the your formulas in the row corresponding to the blank cell at the top of the data column, and starting in the next column over. In our example, if you had five formulas you would enter them in cells B1, C1, D1, E1, and F1. When entering the formulas, refer to the variable using a cell name. This cell is the input cell, and can be any cell on the spreadsheet that is not part of the scenario table and does not contain a formula. You can also use the blank cell at the top of the input column. For example, you might enter the formula `=A1^2` in cell B1 to find the square of the values in column A.

Note that as a result of Mesa's normal formula computation, the cell you enter the formula in will display the value of the formula with the variable equal to the contents of the input cell.

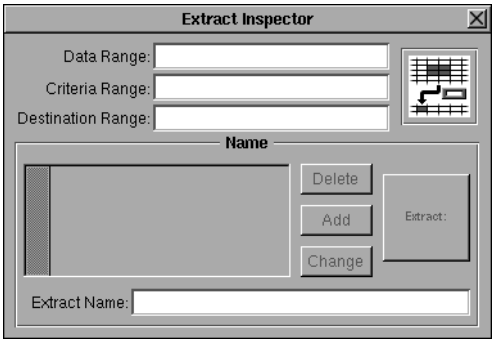
Once you have entered the variable values and the formulas, use the Scenario Inspector to run the scenario. The Scenario Inspector will create a table with the values of the formulas computed for each of the variable's values in the appropriate cell. For example, cell D9 would contain the value of the formula in column D computed with the variable value in row 9. The table is a range on the worksheet that includes the row and column with the input values and formulas. In our example above the range of the scenario would be A1:F11.

Creating Double Input Scenarios

To create a double input scenario, choose a column to enter one set of variables into, and a row to enter the other set of variable. The formula is entered in the cell where the two intersect. For example, if you wanted to place your scenario in the upper left corner of the spreadsheet, you would enter one set of variables in column A starting with cell A2, and one set of variables in row 1 starting with cell B1. The formula would be entered into cell A1. If there were ten values of each variable, the range of this scenario table would be A1:K11. When the scenario is run, a table will be created in the range with the value of the formula using each pair or input values in the appropriate cell. For example, cell C6 would contain the value of the formula using the input values in column C and row 6.

When you enter the formula into the formula cell, refer to the variables using two input cells. The two input cells may be any two cells on the worksheet that are not part of the scenario range and do not contain formulas. As long as these two conditions are met, the input cells can be anywhere on the sheet, and do not have to be blank cells. Once you have entered both the input values and the formula use the Scenario Inspector to run the scenario.

Extract Inspector



This Inspector allows you to extract specific information from a database and put it elsewhere in the worksheet. The icon button can be *Command-dragged* to a Tool Holder or to one of the spots on the Toolbar.

Criteria Ranges

Criteria ranges must contain at least two rows. The top row contains the names of the fields in which to search, and the other rows list the conditions that must be met. Criteria may be numbers, strings, formulas, etc.

	A	B	C	D	E	F	G
1	ID	Name	Region	Sales		Region	Sales
2	101	John	West	high		West	high
3	102	Ann	South	high		South	
4	103	Barbara	East	low			
5	104	Fred	South	average			
6	105	Joseph	West	low			
7	106	Karen	North	average			

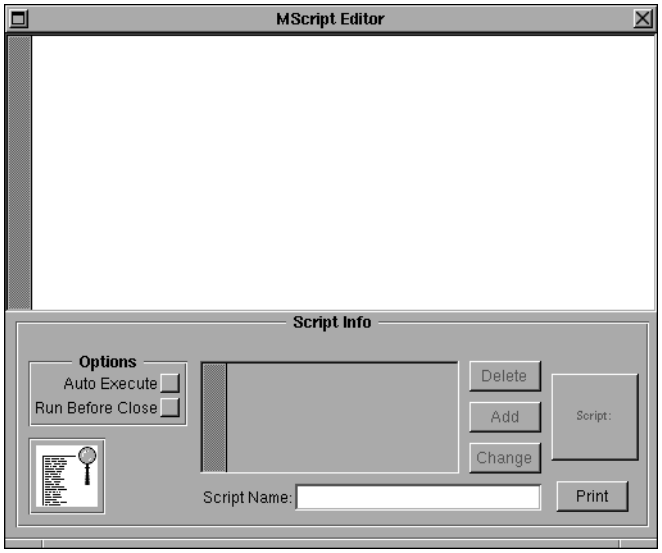
Criteria in the same column are treated as "or" statements, and criteria in different columns are treated as "and" statements. If the criteria range were `f1:f2`, all records whose *Region* field was "West" would be extracted from the database. If the criteria range were `f1:f3`, all records whose *Region* field matched either "West" **or** "South" would be extracted. If the criteria range were `f1:g2`, all records that matched the criteria of *Region* being "West" **and** the *Sales* field being "high" would be extracted.

For criteria containing formulas, the field must be explicitly referred to in the formula. For example, to extract the records for all employees whose ID numbers are greater than 103, the criteria would read `=A2>103`. The same rules apply for "or" and "and" in rows and columns.

Event Inspector

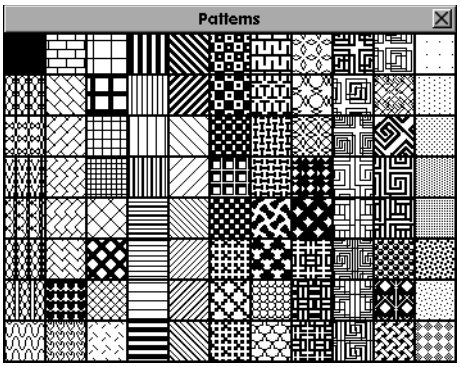
The Event Inspector allows ASCII text to be sent to an Event AddIn. The syntax is the name of the Addin surrounded by first square brackets and then curly braces, the ASCII text, and finally a pipe surrounded by square brackets and then curly braces (i.e. {[|]}). Consult your AddIn manual or your AddIn manufacturer or writer for more information about what must be contained in the ASCII text.

MScript Inspector



The MScript Inspector allows you to edit, print and run MScript scripts. The *Script:* button can be *Command-dragged* to one of the Tool Holders or onto the worksheet. The MScript language is a powerful tool for writing macros and simplifying repetitive tasks. For complete information about the MScript Scripting language, see the *Mesa Programmer's Guide* .

Pattern Panel



The Mesa Pattern panel displays many patterns. Pattern chips can be dragged and dropped from the pattern wells onto cells, ranges, chart elements, reports in the Report Layout Inspector, or any other place that a color can be used.

6. Mesa Menus

Mesa has a hierarchical menu structure consisting of a main menu with subordinate menus, some of which have submenus of their own. A menu item that displays the ▸ symbol opens a submenu. A menu item followed by an ellipsis (…) opens a dialog or inspector panel. If a letter or punctuation mark appears at the far right of the menu item, that command can be executed either by clicking with the mouse on the menu item, or by holding down the *Command* key while typing the indicated character.

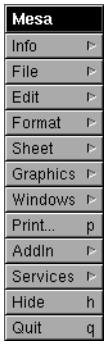
You can click-drag any Mesa menu from its point of origin to any convenient location on your screen. Just point to the menu's title bar, press the mouse button, and drag the menu to a new location. Release the mouse button and the menu stays where you put it.

This feature allows you to put a frequently used menu in a handy spot on your screen and continue to use it while you work. All of the choices on relocated menus will continue to perform as usual. Items can be *Command-dragged* from a menu and placed on the ToolBar or in a tool holder on the worksheet window. (See "ToolBar" on page 158.)

If you have relocated a menu to another place on the screen, an X appears in the upper-right corner of the menu. This is a *close box* . All the menu windows in the illustration (except the Mesa main menu) display close boxes.

You can close a relocated menu by clicking its close box, just as you would with any other window. To close a menu that has not been relocated, just click its name on the parent menu and the submenu will disappear.

The Main Menu



Mesa's menus are organized in a hierarchy starting with the main menu, which is illustrated at left. The main menu contains three commands that you can execute (Print, Hide, and Quit) and a series of submenus that contain combinations of commands, dialog panels, and additional submenus.

A menu item that does not open a submenu is an executable command. You can execute the command by selecting (clicking on) the menu item. For example, to quit Mesa, click on **Quit**. To hide Mesa without quitting, click on **Hide**. The Hide and Quit commands are explained in detail below.

An executable command with an ellipsis (like **Print...**), will display a dialog or inspector panel when selected. The Print command is also explained below.

Executable commands also have keyboard shortcuts. Note the 'q' to the right of the Quit command. If you type *Command-q* (hold down the *Command* key and press and release the *q* key), Mesa will respond as if you had clicked the Quit command.

To select a submenu, click on a menu item that displays the ▸ symbol. The submenu will appear to the right of the main menu. The Mesa submenus are described on the following pages.

Note: Some menu items are nested within submenus. For example, to load a worksheet from disk, you will be instructed to use the **File** → **Open** command.

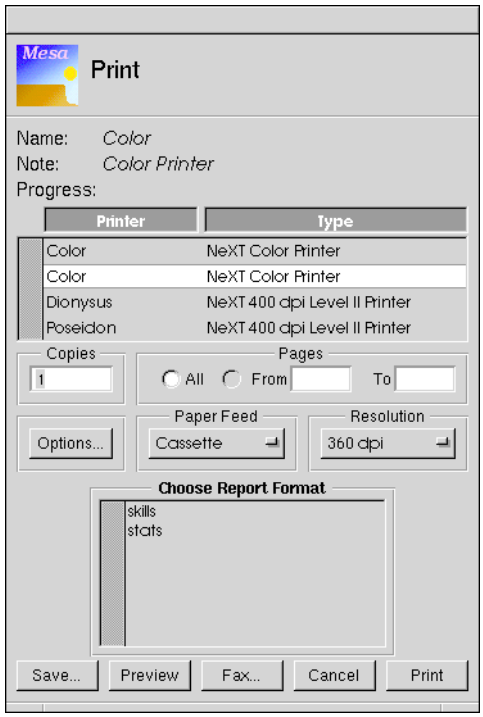
Print

The Print command lets you print all or part of your worksheet or a pre-defined report. You can print your worksheet to a printer, to a fax, , or to the NEXTSTEP Previewer to see how your final output will look.

Select **Print...** from the main menu. Mesa will display the Print dialog panel illustrated below. From the Print panel you can choose to save your output to a PostScript file, preview it, or send it out via fax. If you do not select a pre-defined report to print, Mesa will print the selected range from the currently active worksheet window. If there is no range selected, Mesa will print the whole worksheet.

You can add footers and headers to printouts as well as changing paper size, orientation, and other print parameters via the Page Layout dialog box. (See "Page Layout..." on page 156.)

If you want to print a report that you have defined using the **Format** → **Report Layout...** command, select the report name from the *Choose Report For-*



mat box and select the operation (Save, Preview, Fax, or Print). If you simply double-click on the report format name, Mesa will print that report to the designated printer. For more information on creating reports, see "Report Layout Inspector" on page 102.

The **Print...** command can also be invoked with *Command-p* .

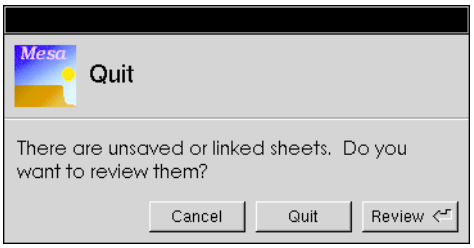
Hide

NEXTSTEP allows you to run more than one program at a time. Unfortunately, the more programs you run, the more cluttered the screen becomes. If you hide an application, it will still continue to run, but none of its windows will appear on the screen. This is useful if you want to work in another pro-

gram, but want to get back to Mesa quickly. To hide Mesa, click on the **Hide** command or type *Command-h*. To unhide Mesa, double-click on the Mesa program icon.

Quit

Quit ends your work session with Mesa. If you have any unsaved worksheets, Mesa gives you a chance to review them and save them before you quit. To quit, select **Quit** from the main menu or type *Command-q* . If you have worksheets that have been changed since the last time you saved them, Mesa will display this dialog panel:



Click on *Review* to individually check each unsaved worksheet before you quit. If you press ↵, Mesa will execute the Review option. If you want to quit from Mesa without reviewing unsaved worksheets, click on *Quit* . If you decide to continue using Mesa, click on *Cancel* and Mesa will continue to run.

The Info Menu

The Info Menu is a submenu of the Main Menu that contains information about Mesa. It is similar to the Info menus you find in most NEXTSTEP programs.

To display the Info Menu, click **Info** on the main menu and the Info Menu will appear to the right of the main menu. The selections on the Info Menu are described below.

About Mesa

See the license number, the version number, and the date of your Mesa program.

Preferences

The Preferences dialog panel lets you set the default spreadsheet size of any new spreadsheets you create by making selections from the Width and Height pop-up lists in the Sheet Size box. Other options let you add or remove add-in programs for spreadsheet functions. The Preferences panel also lets you set units of measurement (inches, centimeters, points), a currency symbol, copy-data formats (EPS, ASCII, tabular), how the *Return, Enter* , and arrow keys will act, and whether or not macros and scripts should autoexecute For more information about the Preferences Panel, see “Preferences Panel” on page 72.

Help

This command invokes the NEXTSTEP help system. The Help window displays a list of topics for which help is available. To get help for a particular menu

item, press the *Alternate* and *Control* keys and click on the menu item for which you desire help. This command can also be invoked with *Command-?* .

Slash Help

Mesa also offers a help facility called Slash-Help that interprets your 1-2-3 slash command and describes how to do the same thing in Mesa. Use **Info → Slash Help** command to access the Slash-Help facility. Then type in your slash command and Mesa will describe the Mesa equivalent of that command. This command can also be invoked with *Command-/* .

Legal

This command opens a scrollable window containing the Mesa License agreement. If you have any questions regarding your legal obligations in using Mesa, please refer to the license. This command can also be invoked with *Command-!* .

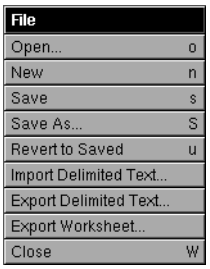
Feedback

The Feedback command is a simple way of sending an e-mail message to Athena Design with questions or comments about Mesa. Please use this facility when you send feedback to us because it includes information about your license number and about which version of *Mesa* you are using.

Show Menus

This option displays all of Mesa's menus and submenus on the screen.

The File Menu



The File Menu is a submenu of the Main Menu. It contains commands relevant to opening, saving, closing, and exporting worksheets.

To display the File Menu, click **File** on the main menu, and the File Menu will appear to the right of the main menu. The selections on the File Menu are described below.

Open

This command displays the NEXTSTEP File Viewer, allowing you to select and open an existing worksheet for use in Mesa. The worksheet file you open can be a Mesa file, or a WKS, WK1, SLK, XLS, or M20 file. This command can also be invoked with *Command-o.*

New

Clicking on this command opens a new, untitled spreadsheet for you to work in. The size of your new spreadsheet is 32x32 cells by default, but you can set a new default in the Preferences panel. (See"Preferences Panel" on page 72) This command can also be invoked with *Command-n* .

Save

Clicking on **Save** will save any changes you have made to the current worksheet without interrupting your work. It is a good idea to use the Save option from time to time to make a permanent copy of your

work. When you save a new, untitled worksheet for the first time, Mesa will prompt you for a file name. This command can also be invoked with *Command-s* .

The first time you save a new worksheet Mesa displays the NEXTSTEP File viewer for you to use. It will ask you to name the worksheet. There will also be a space for you to enter a sheet password. Entering a password will prevent anyone who does not know the password from opening the sheet. If you do not enter a password, the sheet will open normally without asking for a password. The password does not encrypt your sheet; it merely provides a simple form of protection.

Save As...

To save a named worksheet under a new name, use the Save As option. Mesa displays the NEXTSTEP File Viewer for you to use. The Save As command can also be used to save a worksheet using a different password. This command can also be invoked with *Command-Shift-S* .

Revert to Saved

This command restores the form and content of your worksheet as it was the last time you saved it. This command is especially useful if you want to undo the changes you have made since the last save and to start over from a known point. This command can also be invoked with *Command-u* . Note that when you save a worksheet, Mesa automatically creates a backup copy with the same name but with the extension *Mesa~* . You can open this backup file if you need to return to an earlier version of your work.

Import Delimited Text…

Mesa can import many types of delimited text. This command displays the NEXTSTEP File Viewer with different delimiter and end-of-line options, allowing you to select and import an existing file for use in a Mesa worksheet. Possible delimiters include Tab, Comma (,), Space, and Colon (:). End-of-line options include CR, LF, or both. Make sure that the delimiter specifications in the open panel match the delimiters in the ASCII file that you are importing. The data you import will be placed in the pre-selected cell or cell range on your worksheet.

Export Delimited Text…

Mesa displays the NEXTSTEP File Viewer, allowing you to name and save all or part of a worksheet as a delimited text file. If you have selected a range, only the data in that range will be exported. Otherwise, all the data in the worksheet will be exported. The options for exporting text are the same as for importing, listed above.

Export Worksheet...

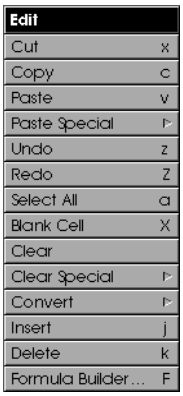
This command allows you to export your worksheet in as either a .wk1 (Lotus) or SYLK worksheet. Many *Mesa* -only functions can not be exported to these formats. Cells containing these functions will be exported as values.

Close

When you are done using a worksheet, choose the **Close** command to remove it from the screen. If you have made any changes since you last saved the

worksheet, Mesa will prompt you to Save it before it is closed. This command can also be invoked with *Command-Shift-W* .

The Edit Menu



The Edit Menu is a submenu of the Main Menu. To display the Edit Menu, click **Edit** on the main menu.

A word about the NEXTSTEP pasteboard: In the NEXTSTEP environment you move or copy information from one location to another by placing it on the *pasteboard* , which is the computer's holding area for text or other information being transferred from one location to another. In other computer environments this tool is often called a *clipboard* . The attributes and uses of the NEXTSTEP pasteboard are fully described in the NEXTSTEP *User 's Reference,* in Chapter 5, *Editing T ext.*

Once information is on the pasteboard, it can be inserted elsewhere in the same window, in another window in the same application, or in a window in a different application.

The Cut, Copy, and Paste commands on the Edit Menu make use of the NEXTSTEP pasteboard. This allows you to copy and paste a selected range from a worksheet to a document in another application, or to copy and paste text from a non-Mesa document into one of your worksheets.

The selections on the Edit Menu are described below.

Cut

This command removes a selected range from your worksheet and stores it in the NEXTSTEP pasteboard. You can enter the contents of the pasteboard somewhere else on the same worksheet, or on another worksheet, by positioning the cursor and using the Paste command. Mesa copies the range of cells to the Pasteboard in a number of different formats as defined in the Preferences panel. (See "Preferences Panel" on page 72.) The Cut command can also be invoked with *Command-x* .

Note: The Move operation is different from a **Cut** and **Paste** operation. If you Cut and Paste a range of cells, formulas that referred to the old range continue to refer that range. If you **Copy** and **Paste Special** → **Move** the cells, formulas in your worksheet are updated to reflect the new location of the range.

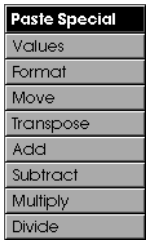
Copy

This command copies a selected range from your worksheet and stores it in the NEXTSTEP pasteboard. You can enter the contents of the pasteboard somewhere else on the same worksheet, or on another worksheet, by positioning the cursor and using the Paste command. Mesa copies the range of cells to the Pasteboard in a number of different formats as defined in the Preferences panel. (See "Preferences Panel" on page 72) The Copy command can also be invoked with *Command-c* . **Copy** copies live link data to the Pasteboard for apps that support live links.

Paste

Use the Paste command to place the contents of the NEXTSTEP pasteboard wherever you want on any worksheet. Just position your cursor where you want the data to appear and click the **Paste** command. Note that the current contents of the pasteboard remain unchanged until overwritten by a new Cut or Copy command. You can Paste the current contents of the pasteboard over and over again. If the Pasteboard contains a TIFF or EPS image, Mesa will place that graphic image on your sheet. You can move and resize the graphic image. Placing graphics in this way allows you to create presentation-quality worksheets. You can also place graphic images on a worksheet by dropping a graphic file on top of the worksheet from the Workspace Manager. Mesa accepts TIFF, EPS, Adobe Illustrator, and Equation Builder files. The Paste command can be invoked with *Command-v* .

Paste Special submenu



The Paste Special submenu has the following commands:

- ***Paste V alues*** lets you paste the values from the cell range that was cut or copied into the pasteboard to another range without affecting its format or content. Note that the values in the pasted range will be frozen and formulas will be converted to their calculated values.
- ***Paste Format*** lets you paste the format of the cell range that was cut or copied into the pasteboard to a specified location on a worksheet without pasting any included values.
- ***Move*** lets you paste the contents of the cell range that was cut or copied into the pasteboard to a spec-

ified location on a worksheet. If the original range was copied to the pasteboard, this command clears the original range as it moves the data to the new one. Any references within the range to cells outside of the range still point to the same place.

Note: You cannot Move ranges from one worksheet to another . To move a range from one worksheet to another , select the range in the first worksheet, **Edit** → **Cut** it, and **Edit** → **Paste** it into the second worksheet.

Note: The Move operation is different from a Cut and Paste operation. If you Cut and Paste a range of cells, formulas that referred to the old location of the range continue to refer that location. If you Copy and Paste Special - Move the cells, formulas elsewhere in your worksheet are updated to reflect the new location of the range.

- Transpose places a copy of the cells on the clipboard at the current location on the worksheet, but transposes the rows and columns of the range. The cell in the second row and first column in the original range is placed in the first row and second column, etc.
- Add adds the value in each cell on the pasteboard to the values in each cell of the selected range.
- Subtract subtracts the value in each cell on the pasteboard from the values in each cell of the selected range.

- Multiply multiplies the values in the selected range by the value of each cell on the pasteboard.
- Divide divides values in each cell of the selected range by the value on the pasteboard.

Undo

Undo causes any edits that you make to the text in the current cell to revert to the original contents. To undo a *series* of actions, you must use the **File** → **Revert to Saved** command. (See "Revert to Saved" on page 140.) This command can be invoked with *Command-z* .

Redo

Redo reinstates the changes you undid with the undo command. It can be invoked with *Command-Shift-Z* .

Select All

The Select All command is used to select all the cells in your worksheet prior to performing an action that affects all cells. You can accomplish the same thing by clicking the empty box in the upper left corner of the worksheet (just above row 1 and to the left of column A) or with *Command-a* .

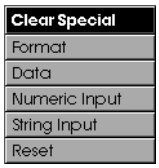
Blank Cell

This command clears the current cell. If you want to blank a cell that is part of a range without affecting any of the other cells in the range, use this command. It can be invoked with *Command-Shift-X*

Clear

The Clear command clears the cell or cell range you specify of both data and formatting information. First, select the cell or range, then click **Clear**. If a graph is selected, Clear will remove it from the worksheet.

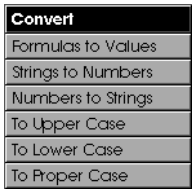
Clear Special submenu



The Clear Special submenu has the following commands:

- ***Format*** clears all formatting options from the cell or cell range that you selected, returning it to Base Style 1.
- ***Data*** clears all data from the cell or cell range that you selected, leaving only the formatting options.
- ***Numeric Input*** clears all cells containing numeric data in the selected cell range. This is useful for worksheets that are used repeatedly , and must be cleared of all prior numeric content before each use. Formatting options, formulas, and string are not affected by this command.
- ***String Input*** clears only cells containing string data in the selected cell range. This command is useful for reusing a worksheet that needs to have only its captions updated. Formatting options, formulas, and number are not affected by this command.
- **Reset** clears the values of cells containing formulas. This is useful when working with cells containing CURRENTV ALUE and CURRENTNUMBER constants.

Convert submenu



The Convert submenu has the following commands:

- ***Formulas to V alues*** freezes the current values in the cell range that you selected. If you recalculate the worksheet, the frozen values will not change.
- ***Strings to Numbers*** converts the string values in a given range to their number value. If a string cannot be converted, the cell is set to 0.
- ***Numbers to Strings*** converts numeric entries in a range to their string value.
- ***To Upper Case*** converts string data in the cell or cell range that you selected to all capital letters.
- ***To Lower Case*** converts string data in the selected cell or cell range to all lower case letters.
- ***To Proper Case*** capitalizes the first letter of each word of string data in the selected cell or cell range

Insert

Use the Insert command to add one or more blank cells, rows, or columns wherever you want on any worksheet. It can be invoked with *Command-j* .

To insert blank columns, highlight one or more columns where you want them to appear and click the Insert command. The number of columns you highlighted will be inserted and all other columns will be shifted to the right. You can then enter or paste information in the new columns you created.

To insert blank rows, highlight one or more rows wherever you need them and click the Insert command. The blank rows will be inserted and all data below the insertion point will shift down.

If you highlight one or more cells on your worksheet and click Insert, the data in the affected row(s) will shift to the right by however many cells you highlighted.

Delete

Delete removes the selected rows, columns or cells from the worksheet and collapses the worksheet appropriately. It can be invoked with *Command-k* .

To delete columns, highlight the columns you wish to delete, and click the Delete command. Columns to the right of the deleted columns will be shifted to the left appropriately.

To delete rows, highlight the rows you wish to delete, and click the Delete command. The Rows below the deleted rows will be shifted up appropriately.

If you highlight one or more cells on your worksheet and click Delete, the data in the affected row(s) will shift to the left by the number of cells you highlighted.

Formula Builder

This command displays an inspector panel containing a scrollable list of all of the formula functions available in Mesa. A pop-up menu lets you choose among the following categories: math, financial, string, constants, special, date, database, and operators. (You can select functions either with or without the @ symbol.)

Formulas beginning with the @ character can be pasted directly into the worksheet, but to use formulas without the @, type "=" in the data input window

of your worksheet, and then define the formula by selecting and pasting non-@ functions from the Formula Builder lists. You must type in any variables required by a function before you can go on.

You can paste operators from the Formula Builder *Operators* list into either @ formulas or non-@ formulas

The Formula Builder can also be invoked with *Command-Shift-F* .

The Format Menu

The Format Menu is a submenu of the Main Menu.

To display the Format Menu, click **Format** on the main menu. The Format Menu appears to the right of the main menu.

A word about inspector panels: The Format Menu introduces two *inspector panels* that let you apply certain options to your cell and sheet formats. Inspector Panels can remain open on your screen (active or inactive) while you continue your work. (See "Cell Style Inspector" on page 75 and "Sheet Inspector" on page 83.)

The selections on the Format Menu are described below.

Cell Style Inspector

If you highlight a cell or cell range and select this command, you will see an inspector panel that shows you the current cell style and allows you to alter it. It is easy to experiment with alternative style settings,

since each change you make in the inspector panel is instantly reflected on the worksheet. If you make a change you don't like, you can change it back right away. When you have made all the cell style changes you want, simply click on the panel's close box and return to your worksheet.

The Cell Style Inspector lets you control style templates, number display formats, decimal places, color, patterns, data alignment, fonts, borders, and input type. If it is more convenient, you can leave the inspector panel on the screen while you continue to work. (See "Cell Style Inspector" on page 75, and "Sheet Inspector" on page 83)

Both the Cell Style and Sheet Inspector panels can remain on your screen while you work.

Cell Format submenu

The Cell Format submenu has eleven commands. To change cell formats, first select a cell or cell range and then click on the desired command on the submenu. Some of these commands are *toggles* . One click creates an effect; the next click reverses it.

You have the following choices:

- ***Bold*** (*Command-b*) toggles the data in the cell range between bold and normal (plain).
- ***Italic*** (*Command-i*) toggles the data in the cell range between italics and plain.
- ***Plain*** restores data in the cell range to un-bold and un-italic.
- ***Bigger Font*** (*Command->*) gradually enlarges the size of the font in the cell range.
- ***Smaller Font*** (*Command-<*) gradually reduces the size of the font in the cell range.

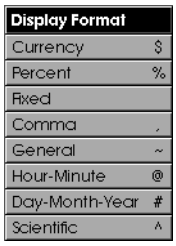
- ***Center*** aligns the data in all selected cells in the center of each cell.
- ***Fill*** fills each cell in the range by repeating its data until the cell is full. You can reverse the effects of the Fill command with the Smart command, described below.
- ***Right*** (*Command-r*) aligns data in all selected cells at the right side of each cell.
- ***Left*** (*Command-l*) aligns data in all selected cells at the left side of each cell.
- ***Smart*** aligns data in all selected cells based on the data type. String data is aligned left, and numeric data is aligned right.
- **Wrap** wraps string text within a cell rather than spilling it over to adjoining cells.

Note that all of the above commands are also available on the Cell Style Inspector panel. In addition, complete information about the available fonts can be found in the font panel that is viewed by clicking the *Fonts* button in the Font box on the Cell Style Inspector panel.

Cell Colors

The command in the Cell Colors submenu will change the text or background colors of the selected cell or range to one of a pre-selected set. Cell colors can also be set with functions. See “Special Functions” on page 251.

Display Format submenu

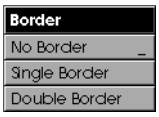


This submenu lets you specify the format of the data within any cell or cell range you specify. Your choices are displayed in the illustration of the menu window above.

- **Currency** (*Command-$*) displays numbers in the default currency format. The default currency format can be changed in the Preferences panel. See "Preferences Panel" on page 72.
- **Percent** (*Command-%*) displays numbers in percentage format with the default number of decimal places.
- **Fixed** displays numbers in fixed decimal format with the default number of decimal places.
- **Comma** (*Command-,*) displays numbers with commas in the appropriate places.
- **General** (*Command-* ;) displays numbers as typed.
- **Hour -Minute** (*Command-@*) displays numbers in HH:MM format.
- **Day-Month-Y ear** (*Command-#*) displays numbers in DD-MM-YY format.
- **Scientific** (*Command-^*) displays numbers in Scientific notation.

All of these choices are also available in the *Display Format* view on the Cell Style Inspector panel. See "Format" on page 77.

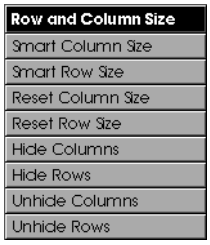
Border submenu



This submenu lets you specify the style of the borders around any cell or cell range you specify. Your choices are displayed in the illustration of the menu window above.

The *Border* box on the Cell Style Inspector panel also offers these choices, in addition to letting you decide on the specific placement of the borders around any cell or range. Cell borders can also be controlled in the Cell Style Inspector. See “Border” on page 82.

Row and Column Size submenu

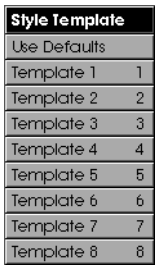


The Row and Column Size submenu has the following commands:

- ***Smart Column Size*** will resize your columns based on the width of the data they contain. This is an especially useful command for reporting purposes, or if you want to display the greatest number of columns within the worksheet window to avoid having to scroll back and forth to see the data.
- ***Smart Row Size*** resizes the worksheet rows based on the height of the data they contain. If you are using large fonts for your strings, you will see the benefit of this command. This works well with the Wrap alignment option.
- ***Reset Column Size*** will resize the columns in your worksheet to their default size if you have altered their size.
- ***Reset Row Size*** will resize the rows in your worksheet to their default size if you have altered their size.
- **Hide Columns** will hide the columns in the selected range. Hidden columns will be skipped when moving around the worksheet with *Tab* or the arrow keys. Column A cannot be hidden.
- **Hide Rows** will hide rows in the selected range. Hidden rows will be skipped when moving around the worksheet with *Tab* or the arrow keys. Row 1 cannot be hidden.

- **Unhide Columns** will unhide columns in the selected range.
- **Unhide Rows** will unhide columns in the selected range.

Style Template submenu



This submenu lets you apply a special style to any cell or cell range you specify. Template styles are defined using the Sheet Inspector Panel in the box labelled *Define T emplates.* (See "Style Templates" on page 76.)

You can define up to eight different template styles, specifying number formats, data alignment, colors, patterns, fonts, and borders. This feature allows you to use special formats to highlight different kinds of data on the same worksheet. You can invoke a specific template style by holding down the *Command* key while typing the number of the template.

The **Use Defaults** option will set a range of cells to the default formatting for its Style Template.

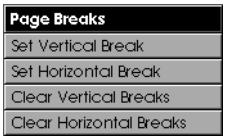
Report Layout...

This command displays the Report Layout Inspector that allows you to define a new report or modify an existing one. You can control cell ranges, margins, tags, and page setup. This panel lists the names of all previously defined reports and lets you add, remove, change, or duplicate any of them. (See "Report Layout Inspector" on page 102.) The Report Layout Inspector can also be invoked with *Command-Shift-R*.

Page Layout...

The Page Layout dialog allows you to choose the orientation of the printed page, choose the paper size, set the margins, set headers and footers, and other parameters that affect printing. Settings in the Page Layout dialog do not affect report formats. For more control over printed reports, use the Report Layout Inspector. (See "Report Layout Inspector" on page 102.) This dialog can also be invoked with *Command-Shift-P* .

Page Breaks submenu



This submenu allows you to set page breaks for use in printing. For more control over printed reports, use the Report Layout Inspector. (See "Report Layout Inspector" on page 102.) The Page Breaks submenu contains the following commands:

- **Set V ertical Breaks** sets a vertical page break at the right of the selected range.
- **Set Horizontal Breaks** sets a horizontal page break below the currently selected range.
- **Clear V ertical Breaks** clears vertical page breaks in the selected range.
- **Clear Horizontal Breaks** clears horizontal page breaks in the selected range.

When page breaks are set the row or column separation corresponding to the page break changes from black to white.

Colors...

The Colors command displays the standard NEXTSTEP Color panel. It allows you to make changes in colors or gray scale. The attributes and uses of the Colors from the Color Picker can be Command-dragged, dropped on cells, ranges, and chart elements to change the color of these items.

Patterns...

The Patterns command displays the Mesa Pattern panel, which allows you to change the pattern of a cell's text or background. There are dozens of patterns available, and they can be dropped onto cells, ranges, chart elements, or any other place that a color can be used. For more information on the Mesa Pattern panel, see "Pattern Panel" on page 131.

Shapes

The Shapes command displays the Mesa Shape Panel, which allows you to drag and drop shapes onto your worksheet. The shapes can be moved around the worksheet, and their dimensions can be changed. The available shapes include a rectangle, oval, line, and a rich text box. For more information about the Shapes panel, see "Shape Panel" on page 101.

Fonts

The Fonts command displays the standard NEXTSTEP Font Panel that shows you a scrollable list of all the type families and typefaces available. The panel also has a table of type sizes in points, and a font viewer window (a "font well").

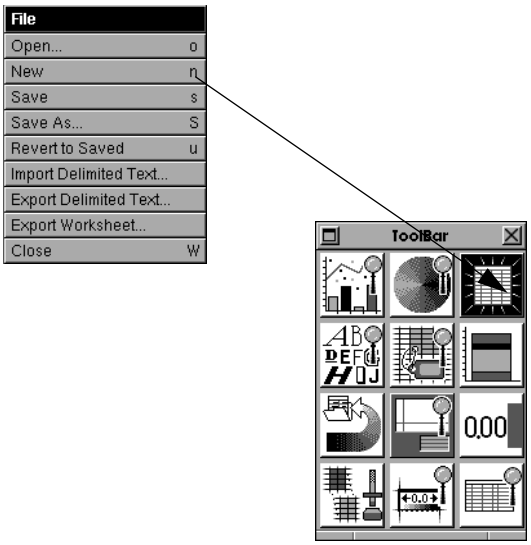
Select a cell or cell range on your worksheet and then click the **Fonts** command. When you choose a type family, a typeface, and a type size in the panel, you can click the *Preview* button to see the result in the font viewer. If you like what you see, click the *Set* button and your font will be applied to the cell range you specified. You can also drag the font from the font well by clicking in the well and dragging the tiny *font chip* icon to the cell or range you specified on your worksheet or drop a font chip on an element in a chart to change its font.

If you change your mind, click *Revert* to restore the original font.

ToolBar

The ToolBar is a very useful feature of Mesa. It allows you to create your own movable window containing a selection of frequently used Mesa commands and have them always ready to use. When you click the ToolBar command, a ToolBar window opens with 12 pre-selected tools and 9 empty spaces, and you can populate it with your own selection of Mesa commands. Do this by pressing the *Command* key while dragging and dropping a command from its regular menu window to a blank square on the ToolBar. The ToolBar can hold up to 21 menu items, macros, reports, extracts, and queries. You can resize the Toolbar if you do not use all of the open squares.

The illustration shows the New command being *Command-dragged* from its menu and dropped into a square on the ToolBar. (The ToolBar in the illustration has been resized to show only 12 squares.) A new tool can be placed on top of an old tool (the old one



will be replaced). To clear an item from the ToolBar, *Command-Control-Click* on the item and release the blank tool onto the tool that you wish to remove.

The Sheet Menu

The Sheet Menu is a submenu of the Main Menu.

To display the Sheet Menu, click **Sheet** on the main menu. The Sheet Menu appears to the right of the main menu.

The selections on the Sheet Menu are described below.

Recalculate

By default, your worksheet will recalculate automatically after every entry. **Recalculate** displays an inspector panel that allows you to turn *Auto Recalc*

on or off. See “Sheet Inspector” on page 83. The Recalculate command is effective only if Auto Recalc is turned off on the Sheet Inspector Panel. After you have made changes to your worksheet, use the Recalculate command (*Command-=*) to recalculate the entire worksheet. If you have more than one worksheet open at the same time, Recalculate will only affect the currently active worksheet. See “Recalc All” on page 160.

Recalc Range

This command is related to Recalculate, but affects only the cell range you specify, plus any cells outside the range that are dependent on the cells within the range. First highlight the range you want recalculated, and then click on this command.

Recalc All

This command is similar to Recalculate, but it will recalculate all currently open worksheets. This can be especially useful for keeping data in linked worksheets up to date.

Goto Cell

The Goto Cell command opens a dialog panel in which you type the address of the cell or the label of the range you wish to see. This command is useful if your worksheet is large enough to be using cells or ranges that are beyond the borders of the worksheet window. (See “Labels submenu” on page 163.) If the name of a graph or chart is entered, the named graph or chart will be selected and displayed. This can also be invoked with *Command-g*.

Run Macro

When you click Run Macro, you will see the Macro Execution dialog panel. Run Macro runs a macro from the current cell. The dialog can also be invoked with *Command-Shift-M* .

Event Inspector

The Event Inspector allows ASCII text to be sent to an Event AddIn. Consult your AddIn manual or your AddIn manufacturer or writer for more information.

MScript Inspector

The MScript Inspector allows you to edit and run MScript scripts. The *Script:* button can be *Command-dragged* to one of the Tool Holders on the worksheet. The MScript language is a powerful tool for writing macros and simplifying repetitive tasks. For complete information about the MScript Scripting language, see *"MScript" on page 306*.

Fill submenu

The Fill submenu has the following commands:

- ***Fill Down*** copies the values from the topmost cell or row down into one or more highlighted cells or rows.
- ***Fill Right*** copies the values from the leftmost cell or column to the right into one or more highlighted cells or columns.
- ***Fill with Cell*** copies the contents of the selected cell into all the cells in the range. This can also be accomplished with *Alternate-* ↵.

- ***Fill Smart*** evaluates the contents of the selected cell before taking any action. If the selected cell contains a value (numeric or string) all highlighted cells are filled with =NEXT(cell), where cell refers to the cell immediately above or to the left. However, if the selected cell contains a formula, all highlighted cells are filled with =SAME(cell), thus accomplishing a run-time copy/paste of the formula. (See "NEXT or @NEXT" on page 245 and "SAME or @SAME" on page 271.) This command can be invoked by pressing *Shift* -↵. The selected cell should be the upper right cell in the range.
- **Fill Smart Down** performs a SmartFill except that it bases the cells in each column of the range on the cell at the top of the column.
- **Fill Smart Right** performs a SmartFill except that is bases the cells in each row of the range on the cell in the first column of the row .

Smart submenu

The Smart submenu has the following commands:

- ***SmartSum:*** (*Command-&*) Highlight a column or row containing numbers you want totalled and extend the highlight to an empty cell at the end. Click **Smartsum**, and the total will automatically appear in the empty cell at the end of the row or column.
- ***SmartMin:*** Highlight a column or row containing numbers and extend the highlight to an empty cell at the end. Click **SmartMin** and the value of the smallest number will automatically appear in the empty cell at the end of the row or column.
- ***SmartMax:*** Highlight a column or row containing numbers and extend the highlight to an empty cell at the end. Click **SmartMax** and the value of the largest number will automatically appear in the empty cell at the end of the row or column.

- ***SmartA verage:*** Highlight a column or row containing numbers and extend the highlight to an empty cell at the end. Click **SmartAverage** and the average value of the numbers will automatically appear in the empty cell at the end of the row or column.
- ***SmartStd Dev:*** Highlight a column or row containing numbers and extend the highlight to an empty cell at the end. Click **SmartStd Dev** and the standard deviation of the numbers will automatically appear in the empty cell at the end of the row or column.
- ***SmartV ariance:*** Highlight a column or row containing numbers and extend the highlight to an empty cell at the end. Click **SmartVariance** and the variance of the numbers will automatically appear in the empty cell at the end of the row or column.
- ***SmartCount:*** Highlight a column or row containing numbers and extend the highlight to an empty cell at the end. Click **SmartCount** and a count of the number of items in the row or column will automatically appear in the empty cell at the end of the row or column. Strings will not be counted.

Labels submenu

The Labels submenu has the following commands:

- ***Label Inspector*** (*Command-Shift-L*) displays an inspector panel that lists all the range labels you have assigned in your worksheet. The inspector panel allows you to Goto, Delete, Run a Macro for , or Update any of the labels on the list.

To add a new label, highlight a range on the worksheet and then open the Label Inspector. The new range address will be displayed and the name win-

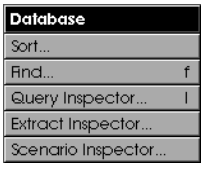
dow will be blank. Type in the name of the new label and click *Add* or press ↵. The label of the new range will be added to the list.

- ***Insert Label Index*** displays a list of all labels and their range addresses in a column on the worksheet, starting in the current cell. This is useful if you frequently refer to the ranges that you have labelled.
- ***Label Range To: Right, Left, Top, Bottom.*** These four commands let you specify the direction of your label range, depending on the label you specify before invoking one of the commands.

If you enter a name in a cell, you can assign it as a range label to any adjacent cell by using the **Sheet → Labels** menu. For example, if you enter *Sales* into a cell and select **Sheet → Labels → Label Range to Right** the label Sales will now refer to the adjacent cell on the right. You can also choose *Left, Top,* or *Bottom* .

- **Delete All Labels** removes all the labels from the worksheet.

Database submenu



The Database submenu has the following commands:

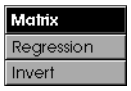
- ***Sort...*** displays a Sort dialog panel that allows you to sort the data on your worksheet. First, select a cell range on your worksheet, then click on the **Sort...** command. You will see a dialog panel where you can choose to sort by rows or columns, and where you can specify up to three sort keys to be used to sort in either ascending or descending order.

If the first row or column of your range contains titles, click the *Has T itles* box to prevent the titles from being sorted.

- **Find...** (*Command-f)* opens the *Find* panel and allows you to search for values and strings in the worksheet.
- **Query Inspector ...** (*Command-Shift-I*) allows you to write SQL queries to retrieve data from a Sybase database and place it into a Mesa worksheet. For more information, see "Query Inspector" on page 115.
- **Extract Inspector ...** allows you to extract specific information from a database and put it elsewhere in the worksheet. See "Extract Inspector" on page 128.
- **Scenario Inspector ...** creates tables which take values for one or two variables, and the result of each value or set of values in a given formula. See "Scenario Inspector" on page 124.

Inspectors submenu

The Inspectors submenu is contains all of Mesa's inspectors. These inspectors can all be found in other places on the menu tree, but the Inspectors submenu provides an easy way to find different inspectors.



Matrix submenu

The Matrix submenu contains Mesa's matrix manipulation commands. These commands can be used to perform various operations on matrices.

- **Regression** performs linear regression on the input range. Place your matrix numbers in a range on the worksheet with the x values in the leftmost columns, and the y value in the second to last col-

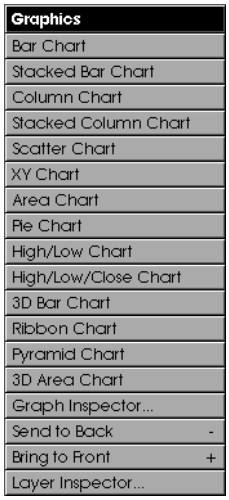
umn, with empty column next to the y values for the output. Highlight the range, and select the command. The results of the linear regression will be calculated and placed in the right-hand column.

- **Invert** will calculate the inverse of a matrix. Highlight the matrix that you wish to find the inverse of, and a range of equal size immediately to the right of it. When you execute the *invert* command, the inverse of the matrix will be placed in the empty cells to the right of the original matrix. Only square matrices may be inverted.

New Window

The New Window command creates an additional window on your current worksheet, with all titles, values, and formulas in place. One or more windows are useful if your worksheet is too large to fit in one window and you frequently need to switch from one part of the worksheet to another. You can easily switch between windows as you work. For your convenience, this command is also included on the Windows submenu. (See "New Window" on page 184.) A new window can also be opened with *Command-Shift-N* .

The Graphics Menu



The Graphics Menu is a submenu of the Main Menu that contains the commands to create each type of graph available in Mesa, and other commands related to working with graphs. To display the Graphics Menu, click **Graphics** on the main menu. The Graphics Menu appears to the right of the main menu.

Once you have created a graph using the Graphics Menu, you can change it from one graph type to another by using the Graph Inspector. Selecting a different option on the graphics menu will create a new graph of the selected type; the original graph will remain unchanged. Double clicking on a graph will open the Graph Inspector.

Each type of chart in Mesa can be modified in a number of ways using the Graph Inspector. You can add titles, labels, and change the orientation and color of the graphs. In addition, some graphs have a *Current* view on the graph inspector that allows you to change the graph in a way specific to that graph type. For more information about graph titles, orientation, color, etc. see “Graph Inspector” on page 88.

Sample Data : The following examples of chart use this data, unless otherwise noted:

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
200	400	600	800	1000	1200	1400	1600

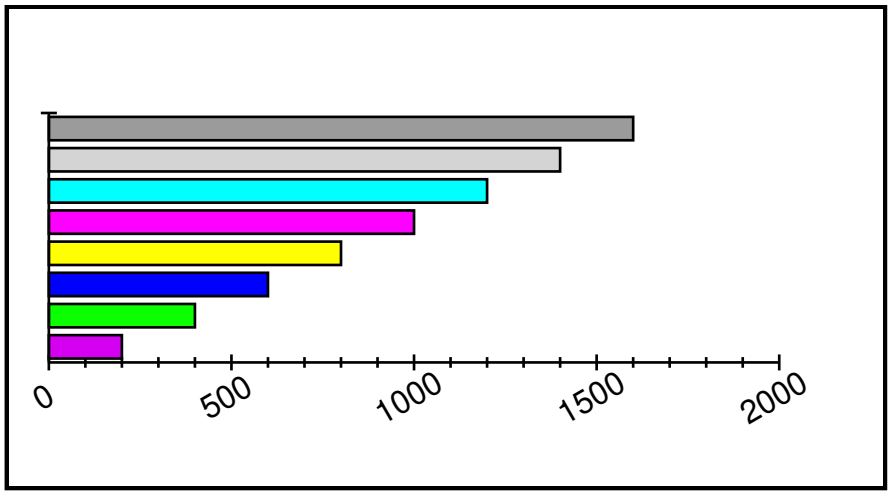
The graph data is set to “Left Column has Titles” off, and Horizontal data orientation. To set these parameters, use the Graph Inspector. To open the Graph Inspector, you can double-click on a graph or select

Graphics → **Graph Inspector...** and then click on the chart that your want to inspect. You can also open the Graph Inspector by double-clicking on a chart. You must have the chart that you want to inspect selected in order to see the Graph Inspector. (See "Graph Inspector" on page 88.)

Note: To remove a chart or graphic from the worksheet, select it and press the ← key or select **Edit** → **Cut**.

Bar Chart

To generate a bar chart, select a cell range and click on the **Bar Chart** command. A bar chart of the selected data will appear on your worksheet. The chart will be in a frame with "handles," so that you can resize the chart and move it out of the way of your work. You can change the color of the bars by



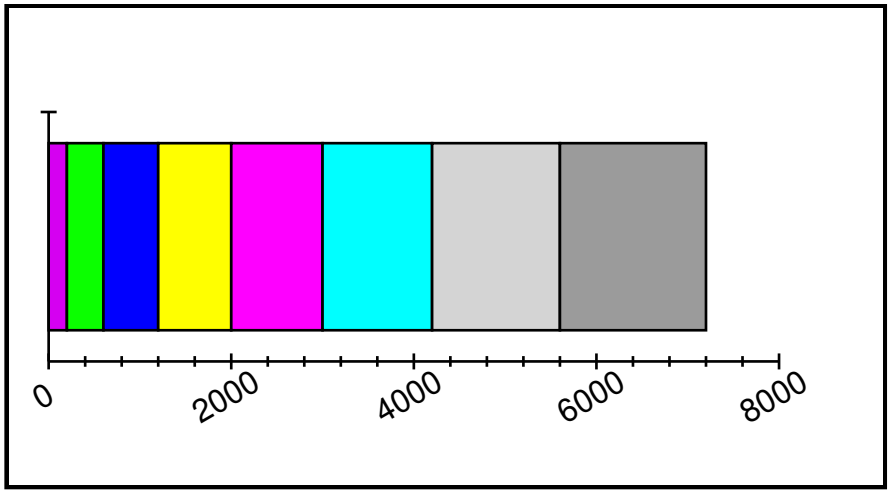
dragging and dropping colors and patterns directly

onto the graph. The colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.

The *Current* view of the Graph Inspector will allow you to adjust the width of the bars. It also allows you to make the first set of data (either the top row or left column depending on the orientation) into a line across the graph.

Stacked Bar Chart

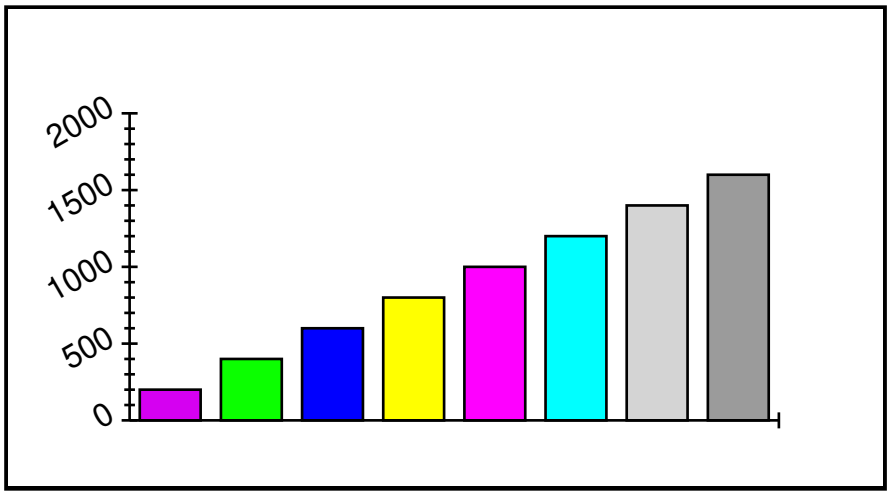
This command generates a stacked bar chart from the data in the range you select. You can change the color of the bars by dragging and dropping colors and patterns directly onto the graph. The colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.



The *Current* view of the Graph Inspector will allow you to adjust the width of the bars. It also allows you to make the first set of data (either the top row or left column depending on the orientation) into a line across the graph.

Column Chart

This command generates a column chart from the data in the range or ranges you select. You can change the color of the columns by dragging and dropping colors and patterns directly onto the graph. The colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.

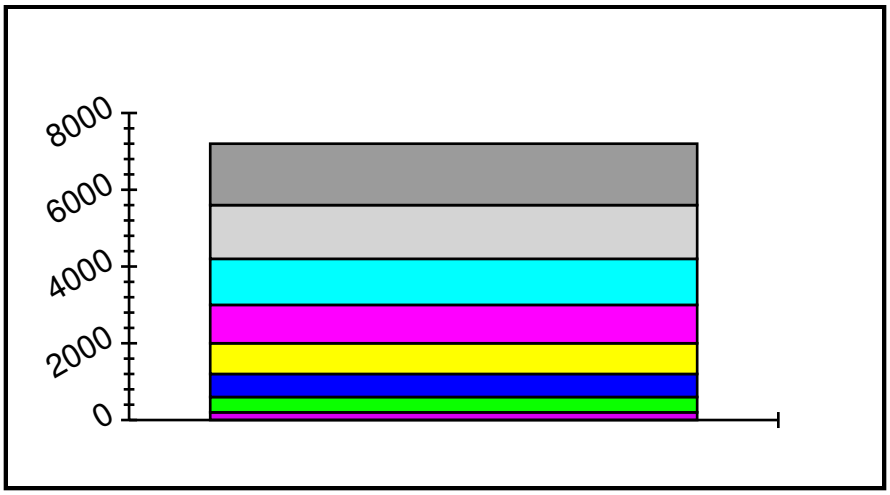


The *Current* view of the Graph Inspector will allow you to adjust the width of the columns. It also allows you to make the first set of data (either the top row or left column depending on the orientation) into a line across the graph.

Stacked Column Chart

This command generates a stacked column chart from the data in the range or ranges you select. You can change the color of the columns by dragging and dropping colors and patterns directly onto the graph.

The colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.

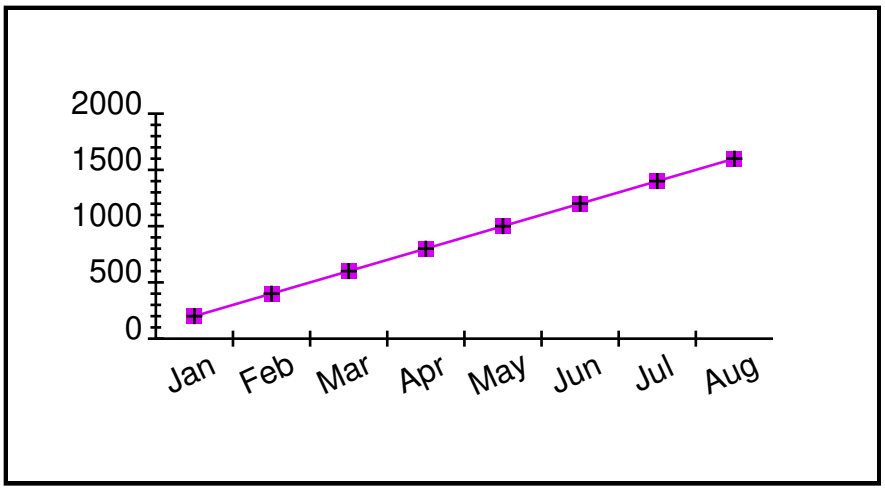


The *Current* view of the Graph Inspector will allow you to adjust the width of the columns. It also allows you to make the first set of data (either the top row or left column depending on the orientation) into a line across the graph.

Scatter Chart

This command generates a scatter chart from the data in the range you select. The colors of the dots will vary to distinguish the data in the cells. You can change the color of the lines and points by dragging and dropping colors and patterns directly onto the

graph. The colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.

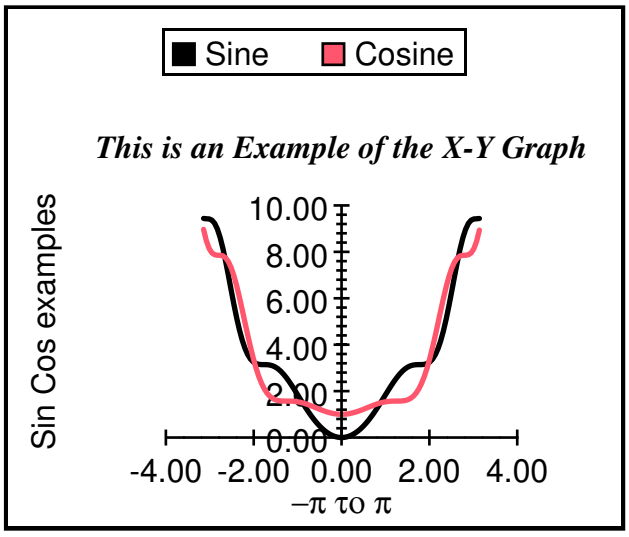


The *Current* view of the Graph Inspector allows you to choose to create scatter charts with points only, lines only, or points and lines. You can also adjust the width of the line.

XY Chart

This command generates a series chart from the data in the range you select. The shapes and colors of the dots will vary to distinguish the data in the cells. You can change the color of the lines by dragging and dropping colors and patterns directly onto the graph.

The colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.

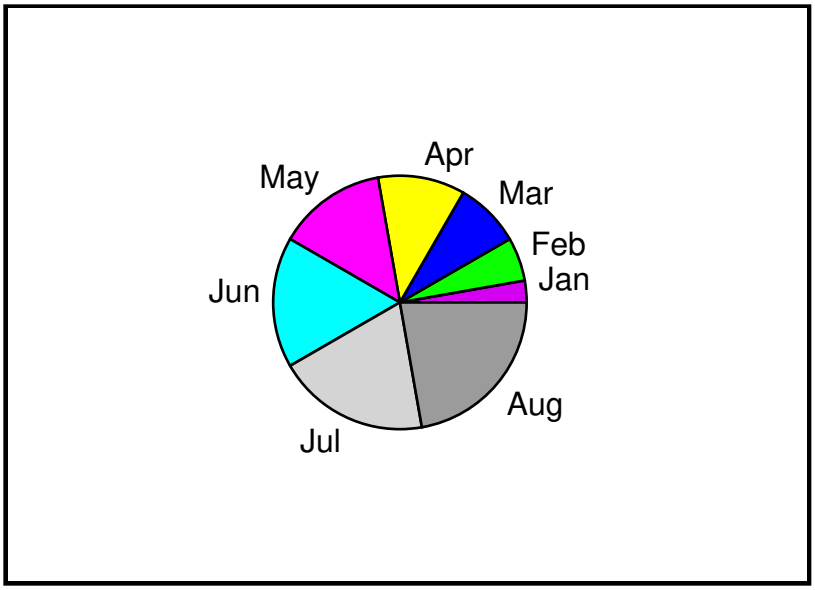


The *Current* view of the Graph Inspector can be used to choose XY graphs with points only, lines only, or points and lines. You can adjust the width of the lines, and you can designate whether there is one x axis, or multiple x axes.

Pie Chart

This command generates a pie chart from the data in the range you select. The colors of the segments will vary to distinguish the data in the cells. You can change the color of the lines and points by dragging and dropping colors and patterns directly onto the

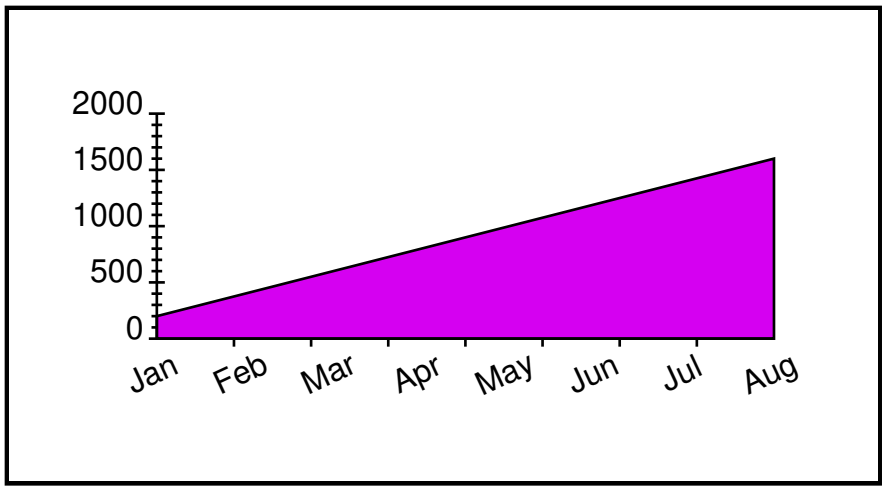
graph. The colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.



If you want to create a pie chart with a portion of the pie offset from the rest, include a range adjacent to the data range with an offset value for each data point. the offset value can have a value from -1 to 1; an offset value of 0 will not move the piece, a positive value will move it away from the center, and a negative value will move it inwards.

Area Chart

This command generates an area chart from the data in the range you select.

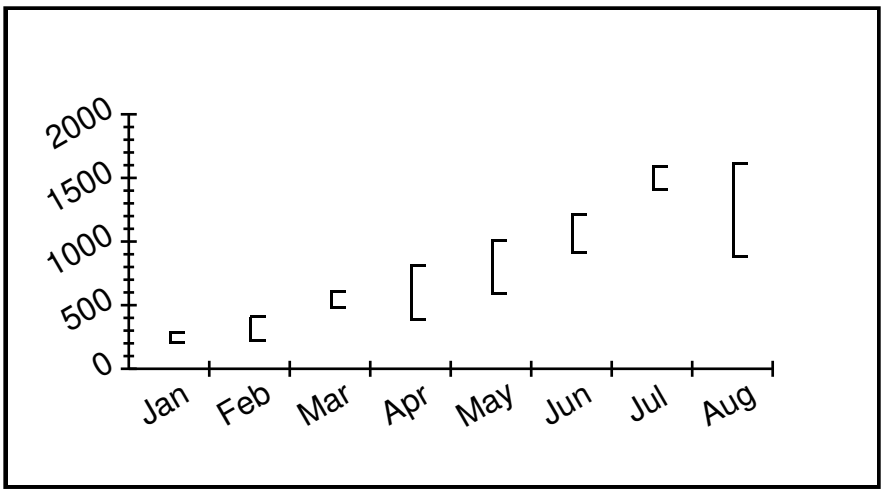


A Change in Data: The High/Low and High/Low/Close charts were generated with the data below. The chart options were set to: Left Column has Titles - off, and Vertical data orientation.

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
200	400	600	800	1000	1200	1400	1600
275	210	475	380	578	904	1582	872
175	470	205	1158	912	1572	1123	1301

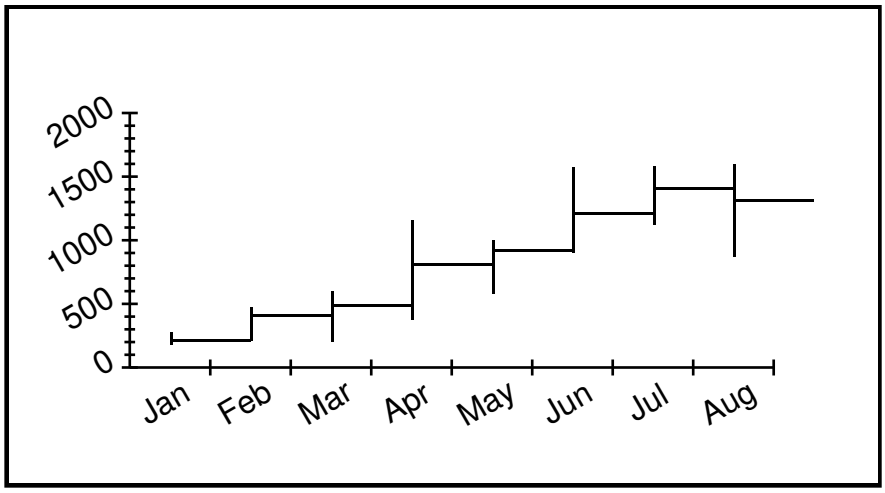
High/Low Chart

This command generates a high/low chart from the data in the range you select.



High/Low/Close Chart

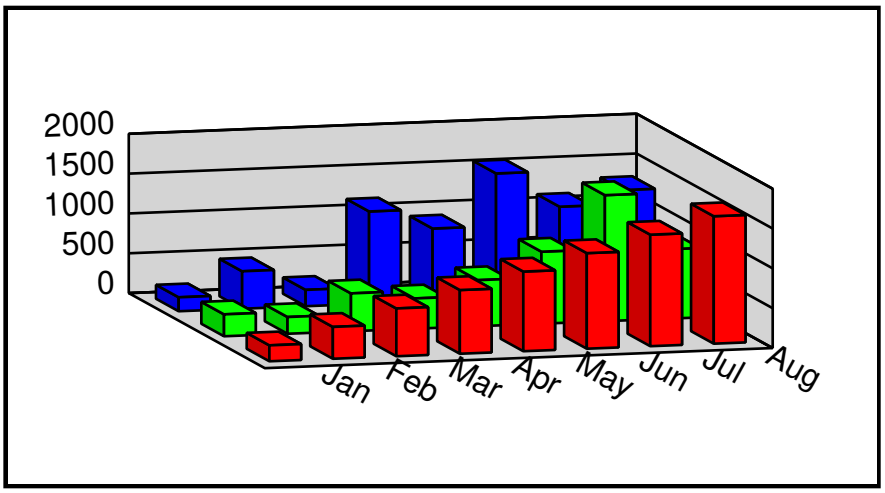
This command generates a high/low/close chart from the data in the range or ranges you select. The default is a candlestick graph, but this can be changed in the *Current* view of the Graph Inspector.



3D Bar Chart

This command generates a 3D Bar Chart from the data in the range or ranges you select. You can change the color of the bars by dragging and dropping colors and patterns directly onto the graph. The

colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.

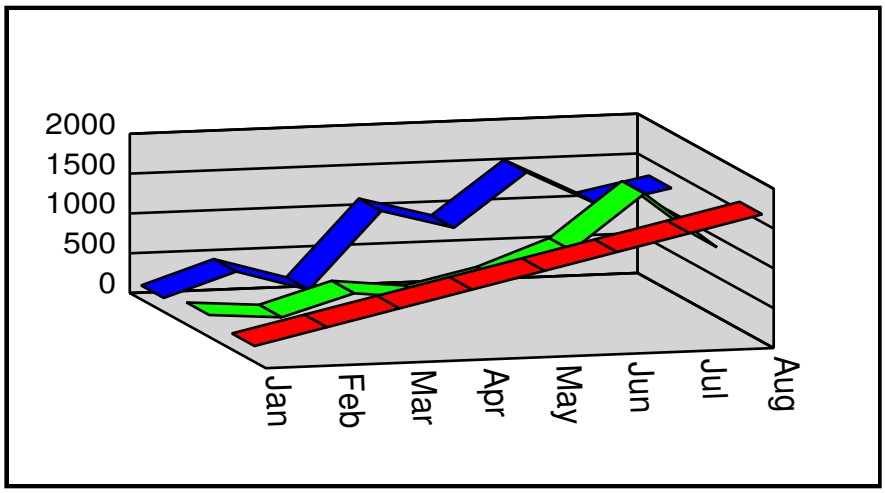


The *Current* view of the Graph Inspector allows you to adjust the X and Y rotation of the graph, so that the graph may be viewed from different angles. There is also a *wire frame* option that allow you to view the bars as wire frames rather than as solid objects.

Ribbon Chart

This command generates a 3D Ribbon Chart from the data in the range or ranges you select. You can change the color of the ribbons by dragging and dropping colors and patterns directly onto the graph. The

colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.

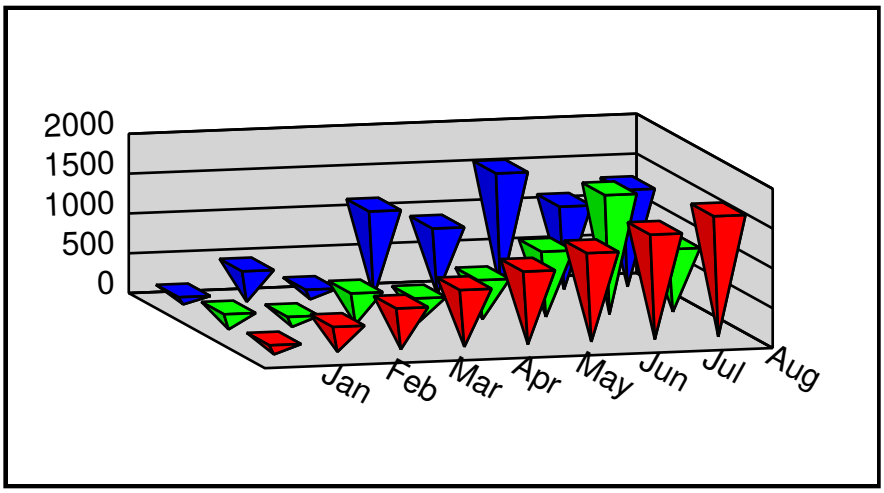


The *Current* view of the Graph Inspector allows you to adjust the X and Y rotation of the graph, so that the graph may be viewed from different angles. There is also a *wire frame* option that allow you to view the ribbons as wire frames rather than as solid objects.

Pyramid Chart

This command generates a 3D Pyramid Chart from the data in the range or ranges you select. You can change the color of the pyramids by dragging and dropping colors and patterns directly onto the graph.

The colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.

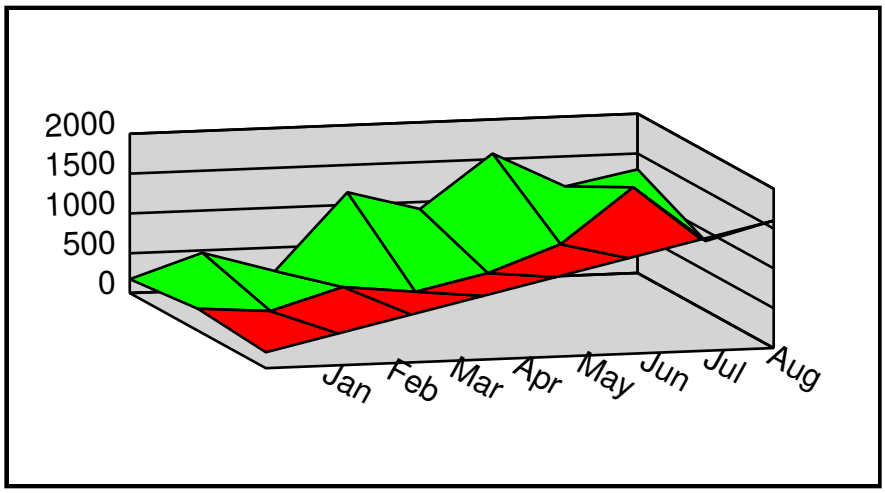


The *Current* view of the Graph Inspector allows you to adjust the X and Y rotation of the graph, so that the graph may be viewed from different angles. There is also a *wire frame* option that allow you to view the pyramids as wire frames rather than as solid objects.

3D Area Chart

This command generates a 3D Area Chart from the data in the range or ranges you select. The data is plotted as the edges of the graph, with the surface representing the area between ranges. You can change the color of the area surface by dragging and dropping colors and patterns directly onto the graph.

The colors and patterns will follow the orientation of the graph, using the same color and pattern for the same set of data.



Graph Inspector

After you have generated a chart, click on it to select it, then select **Graphics → Graph Inspector…** to display the inspector panel. This panel gives you a number of fine-tuning controls for the final appearance of your chart.

Notice the chart-type buttons at the top of the panel. The one that is highlighted when you open the panel represents the type of chart you just created. However, if you click one of the other buttons, you can see how your data looks when displayed on a different chart type. You can do this as often as you like. This is very useful if you are looking for the most effective way to present your data. (See “Graph Inspector” on page 88.)

Send to Back

Use this command if you have two or more charts on your worksheet at the same time to control which one is in full view. To move one chart behind another, click its frame to select it and then click the **Send to Back** command. The selected chart will be placed behind any other charts on the worksheet. This can be invoked with *Command--* (*Command-minus*).

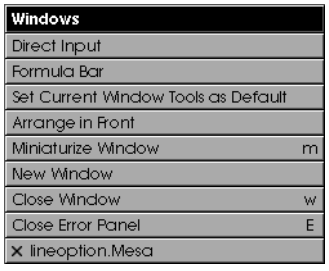
Bring to Front

This is the counterpart of the Send to Back command described above. To bring a chart to the front, click its frame to select it and then click the **Bring to Front** command or use *Command-+* .

Layer Inspector

The Layer Inspector can be used to display graphs or other images above everything, below text but above cell shading, or below everything. TIFF or EPS images can be pasted onto the worksheet, and placed behind the worksheet data. This can be useful for displaying corporate logos and other images on worksheets for presentations.

The Windows Menu



The Windows Menu is a submenu of the Main Menu.

To display the Windows Menu, click **Windows** on the main menu. The Windows Menu appears to the right of the main menu.

The selections on the Windows Menu are described below.

Direct Input

If you select this command while you are working on a spreadsheet, the Formula Bar at the top of the sheet will disappear, allowing you to enter data directly into each cell. To restore the Formula Bar, use the Formula Bar command described below.

Formula Bar

The Formula Bar appears on a newly created spreadsheet by default. If you have been using the Direct Input data entry option described above, you can restore the Formula Bar with this command.

Set Current Window Tools as Default

The worksheet Tool Holders are described in the tutorial section of this book. After you assign tools to the toolboxes, you can have the same tools appear on every new worksheet you create by selecting this command while your worksheet is current.

Arrange in Front

When you have more than one worksheet window displayed on the screen, this command will arrange all the windows neatly in a cascade. The titles of the worksheets appear as items on the Windows menu, so you can easily choose which window you want to be the active one.

Miniaturize Window

Choose this command if you want to "park" a worksheet temporarily in a corner of your screen while you work on a different worksheet. When you select this command or click on the bow in the upper left corner of the worksheet window, the active worksheet turns into a Mesa worksheet icon and moves to the margin of the screen. The icon displays the worksheet title so that you can identify it. You can restore the worksheet to full size by double-clicking its icon.

New Window

The New Window command creates an additional window on your current worksheet, with all titles, values, and formulas in place. One or more windows are useful if your worksheet is too large to fit in one window and you frequently need to switch from one part of the worksheet to another. You can easily switch between windows as you work. Each window in a worksheet can have different icons in its Tool Holders, its own customized home cell, etc.

This command performs the same as **Sheet → New Window**, and is included on this submenu for your convenience. (See "New Window" on page 166.) It can also be invoked with *Command-m* .

Close Window

After you have created a new window on your worksheet, this command lets you save and close the window without closing the worksheet on which it is based. Do not confuse this command with **File →**

Close, which closes the worksheet itself. (See "Close" on page 141.) A window can also be closed with *Command-w* .

Close Error Panel

This command closes the error panel that may appear on your worksheet in response to some unusual condition. The error panel can also be closed with *Command-Shift-E* .

AddIn Menu

This menu will contain any AddIn functions that you have installed. Consult your AddIn manual or your AddIn manufacturer or writer for more information about the menu items that AddIns have installed.

7. Functions

The functions used in formulas provide a way of performing calculations in your worksheet. A formula is a combination of functions, arguments, and operators. Arguments are the input values required for a particular function. Functions perform a calculation and return a value. The result of a function varies, depending on the values and operators you supply for each of the arguments. Formulas containing appropriate functions, arguments and operators can be entered into a cell on the worksheet. When a formula is calculated, the result is displayed in the cell containing the formula. Multiple functions can be used in a single formula.

When working with Mesa's mathematical and constant numbers, keep in mind that because the amount of memory a computer uses to store a number is finite, small inaccuracies can be introduced into the calculations. For example, pi is an irrational number, and while Mesa can calculate it with a high degree of accuracy, it can not calculate it exactly. The formula =SIN(PI) does not return the expected value, 0; it returns a number very close to zero.

This chapter describes how to use Mesa's functions and includes a complete listing of Mesa's operators and functions. Some functions have alternative spellings. When you are given a choice, use whichever spelling you prefer.

Functions and @Functions

In order to provide compatibility with other spreadsheets, Mesa provides functions both with and without a leading @ character. The two sets of functions are very similar, but not always identical, when they are not the same, the @ version of the function will be given a separate entry. Functions without the @ character are compatible with Microsoft Excel functions; function with the leading @ character are compatible with Lotus 1-2-3 functions.

Entering a Formula into a Cell

Mesa recognizes an expression entered into a cell as a formula by the first character of the expression. Formulas must begin with @, =, +, or -. The functions in a formula are assumed to be the non @ character version of the functions unless the @ character is included with each function, even if the formula began with an @ character. This means that in formulas that contain a mixture of @ and non-@ functions, only the functions preceded by an @ character are @ functions.

In Mesa, you can enter a formula into a cell by typing it directly into the formula bar the same way you would enter numbers or strings, or you can build a formula with the Formula Builder.

The Formula Builder is an inspector that contains a scrollable list of all of Mesa's functions, divided up into different categories. Clicking on a function will give you a brief description of the function. The Formula Builder can be used to paste the different functions together. To do this, select the function you want, and click the *Paste* button (for non-@ functions,

type =, +, or - first, if it is the first function in the formula). The highlighted function will appear in the formula bar. Edit any arguments the function has by deleting the variable names from the function as it appears in the formula bar, and replace them with appropriate values. By pasting in operators and other functions, you can create large, complex formulas easily.

Function Arguments

Many of Mesa's functions can take a variety of different arguments. In almost all cases, Mesa's functions can take the specified type of argument, an expression that returns the specified type of argument, or the address of a cell containing the specified type of argument. The following is a brief explanation of what can be used as arguments in Mesa's functions.

Numbers

Many of Mesa's functions take numbers as arguments. The numbers can be entered directly into the formula, or they can be given as the cell address of a cell containing to number. A blank cell has a value of zero.

Occasionally a function needs a specific type of number. Here is a list of some of the specific types of numbers that functions may take.

- **integer** : a number with no fractional portion.
  Ex: 1, 42, -28, 589
- **positive number** : any number with a value of zero or greater .
  Ex: 0, 28, 39.541, 0.028

Strings

When strings are needed as arguments to functions, they can either be typed directly into the formula, in which case they need to be place in quotation marks, or the address of the cell containing the string may be given, in which case the contents of the cell does not have to be in quotation marks. For example, the following expressions both return START.

Examples

```
=UPPER("start")

       A
1      start
=UPPER(A1)
```

Dates

Mesa has a number of date related functions that take dates as their arguments. In most cases these functions take a date serial number as their argument. (Specific exceptions are given in the function descriptions, see "Date Functions" on page 277.)

A date serial number is a number used by Mesa to represent dates so that they may be formatted in a number of different ways. The number corresponds to the number of days since January 1, 1970. Dates after the year 2000 can be specified by placing a 1 before the year . The easiest way to obtain a date serial number is to enter the desired date into a cell as you would normally, and give the address of the cell containing the date.

Cell Addresses

A cell address in Mesa consists of the letter corresponding to the column of the cell, and the number of the row of the cell. For instance, cell A1 is the cell in the upper left hand corner of the worksheet. It is the cell in row one and column A.

Some Mesa functions take a cell address as their argument. In most Mesa functions, when you enter the address of a cell as an argument, the function uses the contents of that cell as the argument. In a few functions, however, the function is looking for information about the cell itself. This is the case for functions that ask specifically for a cell address as an argument.

An example of a function that takes a cell address as its argument is the function ISEMPTY, which checks to see if the cell at a given address is empty.

Examples

```
=ISEMPTY(U2)
```

Ranges

Ranges can also be used as arguments in Mesa's functions. In some cases, the function can take a list of numbers as arguments, and a range that contains an appropriate list may be entered as an argument. In other cases, a function may need to know the size an location of a database, and therefore needs a range as an argument.

Ranges can be specified in one of two ways. The simplest way is to specify it by giving the address of the cells in the upper left corner and lower right corner of the range separated by a colon. For example A1:F10 is the range that occupies rows 1 through 10 of columns A through F. You can also use the name of a ranged that was named using the Label Inspector (See "Label Inspector" on page 114). If you had used the Label Inspector to create a range named Rates, you could enter *Rates* as a range argument.

Examples

```
=ROWS(A1:F10) -> 10
=ROWS(Rates)
```

Constants

Mesa has a number of function constants that can be used as arguments in Mesa's functions. For a complete explanation of Mesa's constants, see "Constants" on page 301.

Expressions

The arguments to functions can contain calls to other functions as well as formulas containing constants and operators. As long as the formulas returns an appropriate value, it may be used as an argument for another function. In this way you can create complex formulas using Mesa's functions.

Operators

! or::

Summary Links worksheets

Usage *sheet* !*adr*

Arguments

sheet	The name of the worksheet to be linked
adr	The address of a cell or range in *sheet*

Description These operators create a link to another worksheet, specified by the filename of the worksheet, and the address of a cell or range in that worksheet. A worksheet link will return the value of the cell or range. Linked worksheets need to be in the same directory .

Examples

```
="Exports"::B52 -> Returns the contents of cell
B52 in the worksheet named Exports.

=U2!I9 -> Returns the contents of cell I9 in the
worksheet whose name is in cell U2 of this work-
sheet

=sum("Exports"!A1:B20) -> Returns the sum of the
numbers in the range A1:B20 in the worksheet named
Exports

=sum(U2::prices) -> Returns the sum of the numbers
in the range labeled prices in the worksheet whose
name is in cell U2 of this worksheet.
```

&

Summary string concatenation

Usage *string1* &*string2*

Description This operator concatenates the two strings on either side of it, forming one string.

Examples

```
="this"&"that" -> thisthat
```

```
        A      B
1      this   that
=A1&B1 -> thisthat
```

= or ==

Summary equal

Usage *exp1* =*exp2*

Description Expressions containing this operator are true if the value of the first expression is equal to the value of the second expression.

!= or <>

Summary not equal to

Usage *exp1* !=*exp2*

Description This operator tests to see if one expression is not equal to another .

Examples

```
if(1!=0, "OK") -> OK
```

<

Summary less than

Usage *num1* <*num2*

Description Expressions containing this operator are true if the value of the first number is smaller than the value of the second number .

>

Summary greater than

Usage *num1* >*num2*

Description Expressions containing this operator are true if the value of the first number is larger than the value of the second number .

<= or =<

Summary less than or equal to

Usage *num1* <=*num2*

Description Expressions containing this operator are true if the value of the first number is either less than or equal to the value of the second number .

>= or =>

Summary greater than or equal to

Usage *num1* >=*num2*

Description Expressions containing this operator are true if the value of the first number is either greater than or equal to the value of the second number .

+

Summary addition operator

Usage *num1* +*num2*

Description This operator returns the sum of the given numbers.

Examples

```
=4+5 -> 9
```

-

Summary subtraction operator

Usage *num1* -*num2*

Description This operator subtracts the second number from the first number .

Examples

```
=10-2 -> 8
```

*****

Summary multiplication operator

Usage *num1 *num2*

Description This operators returns the product of the given numbers.

Examples

```
=3*5 -> 15
```

/

Summary division operator

Usage *num1 /num2*

Description This operator divides the first number by the second number .

Examples

```
=20/5 -> 4
```

---

^

Summary exponentiation operator

Usage *num1 ^num2*

Description This operator raises the first number to the power given by the second number . In other words it multiplies the first number by itself a number of times specified by the second number .

#AND#, &&, <AND>, or AND

Summary logical AND

Usage *exp1* #AND# *exp2*

Description This operator serves as a logical AND. When used to link two expressions, the entire expression is true only if both separate expressions are true.

#OR#, <OR>, OR, or ||

Summary logical OR

Usage *exp1* #OR# *exp2*

Description This operator serves as a logical OR. When used to link two expressions, the entire expression is true if either of the two expressions is true.

<XOR>, XOR, or ^^

Summary exclusive OR

Usage *exp1* <XOR> *exp2*

Description Expressions containing this operator are true if either , but not both, or the given expressions are true.

Math Functions

ABS or @ABS

Summary Computes the absolute value of the given number .

Usage ABS(*num*)

Arguments *num* any number

Description The absolute value of a number is the value of the number ignoring positive or negative indicators. The absolute value of a positive number is given as the number . The absolute value of a negative number would be that number multiplied by negative one (-1); in other words the same number , except positive.

Examples

```
=ABS(-45.3)          -> 45.3
@ABS(-45.3)          -> 45.3
=ABS(COS(PI))        -> 1
=ABS(34)             -> 34
=ABS("Absolutely!")  -> 0
```

ACOS or @ACOS

Summary Returns the arccosine of the given number .

Usage ACOS(*num*)

Arguments *num* any number between -1 and 1

Description The arccosine of the number *num* is the angle whose cosine is *num* . The ACOS function gives the value of the angle in radians; the value of the angle can be

obtained in degrees by using this function in conjunction with the RADTODEG function. See “RADTODEG or @RADTODEG” on page 216.

Examples

```
=ACOS(0.7071) -> 0.7854 (radians) : This is equal
to an angle of 45 degrees.

@RADTODEG(@ACOS(0.7071)) -> 45 : This gives the
same answer as above, but in degrees.
```

ASIN or @ASIN

Summary Returns the arcsine of the given number .

Usage ASIN(*num*)

Arguments *num* any number between -1 and 1

Description The arcsine of the number *num* is the angle whose sine is *num* . The ASIN function gives the value of the angle in radians; the value of the angle can be obtained in degrees by using this function in conjunction with the RADTODEG function. See “RADTODEG or @RADTODEG” on page 216.

Examples

```
=ASIN(0.3090) -> 0.3141 (radians) : This is equal
to an angle of 18 degrees.

@RADTODEG(@ASIN(0.3090)) -> 18 : This gives the
same answer as above, but in degrees.
```

ATAN or @ATAN

Summary Returns the arctangent of the given number .

Usage ATAN(*num*)

Arguments *num* any number

Description The arctangent of the number *num* is the angle whose tangent is *num* . The A TAN function gives the value of the angle in radians; the value of the angle can be obtained in degrees by using this function in conjunction with the RADTODEG function. See "RADTODEG or @RADTODEG" on page 216.

Examples

```
=ATAN(5.6713) -> 1.3963 (radians) : This is equal
to an angle of 80 degrees.
```

ATAN2

Summary Returns the arctangent of the ratio of the given X and Y coordinates.

Usage ATAN2(*x, y*)

Arguments *x* any number
y any number

Description The A TAN2 function finds the angle between the X axis and the point with the given coordinates *x* and *y*. This gives the angle of a line from the origin (0,0) to the point (*x*,*y*). The A TAN2 function gives the value of the angle in radians; the value of the angle can be obtained in degrees by using this function in conjunction with the RADTODEG function. See "RADTODEG or @RADTODEG" on page 216.

Examples

```
=ATAN2(1,3) -> 0.3218 (radians) : This is equal to
an angle of 18.4 degrees.
```

@ATAN2

Summary

Returns the arctangent of the ratio of the given Y and X coordinates.

Usage

@ATAN2(*y, x*)

Arguments

y	any number
x	any number

Description

@ATAN2 finds the angle between the X axis and the point with the given coordinates *x* and *y*. This gives the angle of a line from the origin (0,0) to the point (*x*,*y*). The @ATAN2 function gives the value of the angle in radians; the value of the angle can be obtained in degrees by using this function in conjunction with the RADTODEG function. See "RADTODEG or @RADTODEG" on page 216.

Examples

```
@ATAN2(3,1) -> 0.3217 (radians): This is equiva-
lent to an angle of 18.4 degrees.
```

Note: ATAN2 and @A TAN2 are different functions. ATAN2 requires the x parameter to be first, whereas @A TAN@ requires the y parameter to be first.

AVE or AVERAGE

Summary

Computes the average, or mean, of a list of numeric values. Strings are ignored.

Usage

AVE(*num* , ...), AVE(*range*), AVE(*range1, range2, range3,...*)

Arguments

num any number
range any worksheet range

Description

Divides the sum of the given numbers by the count of numeric entries. The items in the list are separated by commas, and they may be numbers or cell/range references containing numbers. You may list any combination of numbers, cell addresses, and ranges as arguments. String entries are ignored and have no effect on the result.

Examples

```
=AVERAGE(4,5,25,3) -> 9.25

Given the following data:
A1: 184
A2: 592
A3: 97
=AVE(A1:A3) -> 291
```

@AVE or @AVG

Summary

Computes the average, or mean, of a list of numeric values. Strings are treated as zero values.

Usage

@AVE(*num* ,...), @AVE(*range*), @AVE(*range1, range2, range3,...*)

Arguments

num any number
range any worksheet range

Description

Divides the sum of the given numbers by the count of numeric entries. The items in the list are separated by commas, and they may be numbers or cell/range references containing numbers. You may list any combination of numbers, cell addresses, and ranges as arguments. String entries are treated as zero and are computed into the result.

Examples

```
@AVE(4,5,25,3) = 9.25

Given the following data:
A1: 184
A2: 592
A3: 97
@AVE(A1:A3) = 291
```

Note: AVE and AVERAGE are different from @AVE and @AVG in that AVE and AVERAGE ignore string values, while @AVE and @AVG treat string values as having a zero value.

COS or @COS

Summary Returns the cosine of the given number .

Usage COS(*num*)

Arguments *num* any number

Description The cosine of the angle *num* is the X coordinate of the point where a line at that angle from the X axis intersects a circle of radius one (1) centered on the origin (0,0). The COS function takes the value of the angle in radians; the value of the angle can be converted from degrees to radians within this function by using it in conjunction with the DEGTORAD function. See "DEGTORAD or @DEGTORAD" on page 206.

Examples

```
=COS(PI) -> -1
@COS(LN(3)) -> 0.4548
```

COUNT

Summary Counts the number of numeric entries. Cells or items containing blanks or strings are not counted.

Usage COUNT(*num , ...*), COUNT(*range*), COUNT(*range1, range2, ...*)

Arguments *num* any number
range any worksheet range

Description The COUNT function finds the number of numeric entries in the list of arguments, and ignores string values. The items in the list are separated by commas, and they may be numbers or cell/range refer-

ences containing numbers. Y ou may list any combination of numbers, cell addresses, and ranges as arguments.

Examples

```
          A       B
1       Duck      10
2       Soup      20
3          1.86   20
4          $300   10
5       '======   30
6                 50
7             40  70

=COUNT(A1:A7) -> 3 : There are only three items in
column A because cells A1, A2, and A5 are strings
and A6 is blank.

=COUNT(A1:A7,B1:B7) -> 10 : This counts the 3
items in column A and the 7 items in column B.

=COUNT("Good","Bad","Indifferent",0) -> 1
```

@COUNT

Summary Counts the number of numbers and strings given.

Usage @COUNT(*num, ...*), @COUNT(*range*), @COUNT(*range1, range2, ...*)

Arguments

num any number
range any worksheet range

Description The @COUNT function finds the number of numeric or string entries in the list of arguments. Cells or items containing strings are counted. Cells or items

containing blanks are not counted. You may list any combination of numbers, cell addresses, and ranges as arguments.

Examples

```
          A       B
1       Duck      10
2       Soup      20
3          1.86   20
4          $300   10
5       '======   30
6                 50
7             40  70

@COUNT(A1:A7) = 6: There are only six items in
column A because cell A6 is blank.

@COUNT(A1:A7,B1:B7) = 13: This counts the 6 items
in column A and the 7 items in column B.

@COUNT("Good","Bad","Indifferent") = 3
```

Note: COUNT and @COUNT are different in that COUNT only counts numeric values. @COUNT also counts string values.

DEGTORAD or @DEGTORAD

Summary Converts the given number from degrees to radians.

Usage DEGTORAD(*num*)

Arguments *num* any number

Description Angles can be expressed in units of degrees or radians. There are 360 degrees, or 2 π radians in a circle. All of Mesa's trigonometric functions express angles in radians. This function will allow you to convert angles given in degrees to radians for use in Mesa's

trigonometric functions. Radians can be converted to degrees using the RADTODEG function. See "RADTODEG or @RADTODEG" on page 216.

Examples

```
=DEGTORAD(180) -> 3.14159
```

DIV or @DIV

Summary

Divides the given number by the given divisor and returns an integer .

Usage

DIV(*num, div*)

Arguments

num	any number
div	any non-zero number

Description

This function divided the given number by the given divisor and returns the integer portion of the answer . The remainder can be obtained using the MOD function. See "MOD or @MOD" on page 214.

Examples

```
=DIV(12,5) -> 2
```

EXP or @EXP

Summary Raises e the power of the given number .

Usage EXP(*num*)

Arguments *num* any real number

Description EXP raises e to the given power . The number e is defined to be the number such that the derivative of e^{num} is e^{num}. The number e is approximately 2.71828. The natural logarithm of e^{num} is *num.* See "LN or @LN" on page 209.

Examples

```
=EXP(4) -> 54.5982

@EXP(12%*5) * 30000 -> $54,664  :  This formula
calculates the present value of $30,000 invested
at 12% continually compounded interest for a term
of 5 years.
```

FRAC or @FRAC

Summary Returns the fractional component of the given number.

Usage FRAC(*num*)

Arguments *num* any number

Description FRAC returns the non-integer portion of the given number , *num* . This function complements the INT function, see "INT or @INT" on page 209.

Examples

```
=FRAC(NOW)  :  This formula returns only the time
component of the current date.

@FRAC(3.14159) -> 0.14159
```

INT or @INT

Summary Returns the integer portion of the given number .

Usage INT(*num*)

Arguments *num* any number

Description INT returns the integer portion of the given number , *num* . This does not round the number off, it simply ignores all digits to the right of the decimal point. This function complements the FRAC function, see "FRAC or @FRAC" on page 208.

Examples

```
=INT(RAND*10)   :  This formula returns a random
number between 0 and 9.

A3: 68293
@INT(LOG(A3))+1 -> 5  :  This formula tells how
many digits are the number in cell A3.
```

LN or @LN

Summary Returns the natural logarithm of the given number .

Usage LN(*num*)

Arguments *num* any positive number

Description The natural logarithm of a number is the power to which e needs to be raised to result in the given number; in other words, the logarithm to base e. The number e is defined to be the number such that the derivative of e^{num} is e^{num} . The number e is approxi-

mately 2.71828. If the natural logarithm of *num* is raised to the power e, the result is *num.* See "EXP or @EXP" on page 208.

Examples

```
=LN(27) -> 3.2958

B8: 1024
=LN(B8)/LN(2) -> 10  :  This formula returns the
base 2 log of cell B8.
```

LOG, LOG10 or @LOG

Summary Returns the base ten (10) logarithm of given number .

Usage LOG(*num*)

Arguments *num* any positive number

Description The base ten (10) logarithm of a number is the power to which ten needs to be raised in order to result in the given number .

Examples

```
=LOG10(45) -> 1.6532

+10^(LOG(292)) -> 292  :  This formula takes the
base 10 log of 292, then raises 10 to that power,
resulting in the number it began with.
```

MAX

Summary Returns the largest number in the given range or list.

Usage MAX(*num,...*), MAX(*range*), MAX(*range1, range2,...*)

Arguments
num any number
range any worksheet range

Description MAX returns the largest number in a given list or range, ignoring strings and non-numeric input. The items in the list are separated by commas, and they

may be numbers or cell/range references containing numbers. Y ou may list any combination of numbers, cell addresses, and ranges as arguments.

Examples

```
=MAX(4.5,3.2,2.5,2.5,6.2) -> 6.2

=MAX(4,4,4) -> 4

          A     B     C
1         300         400
2           0         400
3         200         100
4         300        5000
5         700
6         600
7         300
8         500
=MAX(A1:A8,C1:C8) -> 5000
```

@MAX

Summary

Returns the largest number in the given range or list with strings considered to be zero.

Usage

@MAX(*num,...*), @MAX(*range*), @MAX(*range1, range2,...*)

Arguments

num any number
range any worksheet range

Description

@MAX returns the largest number in a given list or range. The items in the list are separated by commas, and they may be numbers or cell/range references containing numbers. Y ou may list any

combination of numbers, cell addresses, and ranges as arguments. Strings are treated as having a value of zero.

Examples

```
@MAX(4.5,3.2,2.5,2.5,6.2) = 6.2

@MAX(4,4,4) = 4

          A     B     C
1         300         400
2           0         400
3         200         100
4         300        5000
5         700
6         600
7         300
8         500
@MAX(A1:A8,C1:C8) -> 5000
```

Note: MAX() and @MAX() differ in that MAX() ignores string values, while @MAX treats string values as zero.

MIN

Summary

Results in the smallest number in the given range or list.

Usage

MIN(*num,...*), MIN(*range*), MIN(*range1, range2,...*)

Arguments

num any number
range any worksheet range

Description

MIN returns the smallest number in a given list or range, ignoring strings and non-numeric input. The items in the list are separated by commas, and they

may be numbers or cell/range references containing numbers. Y ou may list any combination of numbers, cell addresses, and ranges as arguments.

Examples

```
A1: 300, A2: 400, A3: Hello!
=MIN(A1:A3) -> 300  :  This formula returns 300,
and the string entry in cell A3 is ignored.

A1: 0.0002, A2: 0, A3: -339492
+MIN(A1:A3) -> -339492
```

@MIN

Summary

Returns the smallest number in the given range or list, with string values considered to be zero.

Usage

@MIN(*num,...*), @MIN(*range*), @MIN(*range1, range2,...*)

Arguments

num any number
range any worksheet range

Description

@MIN returns the smallest number in a given list or range. The items in the list are separated by commas, and they may be numbers or cell/range references containing numbers. Y ou may list any

combination of numbers, cell addresses, and ranges as arguments. Strings are treated as having a value of zero.

Examples

```
A1: 300
A2: 400
A3: Hello!
@MIN(A1:A3)-> 0 :  This formula returns zero
because the string "Hello!" evaluates to zero.

A1: 0.0002
A2: 0
A3: -339492
@MIN(A1:A3) -> -339492
```

Note: MIN() and @MIN() differ in that MIN() ignores string vales while @MIN() treats strings as having a value of zero.

MOD or **@MOD**

Summary Returns the remainder of the given number divided by the given divisor .

Usage MOD(*num, div*)

Arguments

num any number
div any non-zero number

Description MOD compliments the DIV function (see "DIV or @DIV" on page 207) by returning the portion of the given number that does can not be divided evenly by the given divisor .

Examples

```
=MOD(12,5) -> 2

@NOW-@MOD(@NOW,7)+5  :  This function returns the
date-number for Monday of the current week.
```

PROD

Summary Multiplies all the numbers in the given list.

Usage PROD(*num, ...*), PROD(*range*), PROD(*range1, range2, ...*)

Arguments

num any number
range any worksheet range

Description PROD takes the product of all the numbers in the given list. The items in the list are separated by commas, and they may be numbers or cell/range references containing numbers. Y ou may list any combination of numbers, cell addresses, and ranges as arguments. Strings within the list are ignored.

Examples

```
=PROD(3, 5, 10, 2) -> 300

A1:  3
A2:  5
B1: 10
=PROD(A1:A2, B1, 2) -> 300
```

@PROD

Summary Multiplies all the numbers in the given list. Strings are treated as having a value of zero.

Usage PROD(*num, ...*), PROD(*range*), PROD(*range1, range2, ...*)

Arguments

num any number
range any worksheet range

Description PROD takes the product of all the numbers in the given list. The items in the list are separated by commas, and they may be numbers or cell/range references containing numbers. Y ou may list any com-

bination of numbers, cell addresses, and ranges as arguments. Strings within the list are treated as having a value of zero, so if a string appears in the list, PROD will return zero.

Examples

```
@PROD(3, 5, 10, 2) -> 300

A1:  3
A2:  5
B1: 10
@PROD(A1:A2, B1, 2) -> 300
```

Note: PROD and @PROD differ in that PROD ignores strings, while @PROD treats strings as having a value of zero.

RADTODEG or @RADTODEG

Summary Converts the given number from radians to degrees.

Usage RADTODEG(*num*)

Arguments *num* any number

Description Angles can be expressed in units of degrees or radian. There are 360 degrees, or 2 π radians in a circle. All of Mesa' s trigonometric functions express angles in radians. This function will allow you to convert angles given in radians by Mesa' s trigonometric functions to degrees. Degrees can be converted to radians using DEGTORAD. See "DEGTORAD or @DEGTORAD" on page 206.

Examples

```
=RADTODEG(PI) -> 180
```

ROOT or @ROOT

Summary Returns the given root of the given number .

Usage ROOT(*num, n*)

Arguments

num	any positive number
n	any number

Description This function can be used to find the *n*th root of a number , *num* . The *n*th root of a number *num* is the number that, multiplied by itself *n* times, returns the number *num* .

Examples

```
=ROOT(81,4) -> 3
```

ROUND or @ROUND

Summary Rounds a number to a given number of decimal places.

Usage ROUND(*num, prec*)

Arguments

num	any number
prec	any integer

Description This function rounds the number *num* to *prec* decimal places. If *prec* is zero, the number is rounded to the nearest integer . If *prec* is greater that zero, it will round off *prec* places beyond the decimal point. If *prec* is less than zero, it rounds to the left of the decimal place.

Examples

```
=ROUND(350.2852,2) -> 350.29
@ROUND(25492,-3) -> 25000
```

SGN, SIGN, @SIGN, or @SGN

Summary Determines whether a number is positive or negative.

Usage SGN(*num*)

Arguments *num* any number

Description This function returns 0 if *num* is zero, -1 if *num* is less than 0, and 1 if *num* is greater than 0. This functions is useful in conjunction with other functions where it is necessary to determine whether a number is positive, negative, or zero.

Examples

```
B7: +RAND
=SIGN(B7-0.5) : This formula returns -1, 0, or 1
randomly.

=SGN(-405) -> -1
```

SIN or @SIN

Summary Computes the sine of the given number .

Usage SIN(*num*)

Arguments *num* any number

Description The sine of an angle is the Y coordinate of the point where a line at that angle from the X axis intersects a circle of radius one (1) centered on the origin. The SIN function takes the value of the angle in radians; the value of the angle can be converted from degrees

to radians within this function by using it in conjunction with the DEGTORAD function. See "DEGTORAD or @DEGTORAD" on page 206.

Examples

```
=SIN(60*PI/180) = 0.8660 : This formula gives the
sine of 60 degrees.

A mountain road goes up at an incline of 25
degrees.  If the road is straight and is 20 miles
long, what vertical distance will a car travel to
climb it? The formula is +20 * SIN(25*(pi/180)) ->
8.45 miles
```

SQRT or @SQRT

Summary Calculates the square root of a number .

Usage SQRT(*num*)

Arguments *num* any positive number

Description This function finds the positive number which, when multiplied by itself, yields the number *num* . To find roots other than the square root, see "ROOT or @ROOT" on page 217.

Examples

```
=SQRT(34) -> 5.8310
@SQRT(@LOG(200)) -> 1.5169
=SQRT(VAR(200,500,100)) -> 170 : This formula com-
putes the standard deviation of the values given
in the variance function.
```

STDDEV or STDEV

Summary

Returns the standard deviation of the ranges or list. Blank cells and strings are not counted.

Usage

STDDEV(*num,...*), STDDEV(*range*), STDDEV(*range1, range2,...*)

Arguments

num any number
range any worksheet range.

Description

This function finds the standard deviation of the given range or list of numbers. The items in the list are separated by commas, and they may be numbers or cell or range references containing numbers. You may list any combination of numbers, cell addresses, and ranges as arguments. The standard deviation is a measure of the statistical distribution of a set of numbers.

Examples

```
          A       B
1       NAME    SCORE
2       Anna     65.8
3       Bill     95.4
4       Donna    30.2
5       Mark     54.9
6       Maria    35.1
7       Susan    75.9
8       John     83.2
9       Rob      33.1
10      Ethan    81.8

=STDEV(B2:B10) -> 23.1
```

@STD

Summary

Returns the standard deviation of the ranges or list. Strings are counted as 0, and blank cells are not counted.

Usage

@STD(*num,...*), @STD(*range*), @STD(*range1, range2,...*)

Arguments

num any number
range any worksheet range

Description

This function finds the standard deviation of the given range or list of numbers. The items in the list are separated by commas, and they may be numbers or cell or range references containing numbers. Y ou may list any combination of numbers, cell addresses, and ranges as arguments. Strings within the list are treated as having a value of zero. The standard deviation is a measure of the statistical distribution of a set of numbers.

Examples

```
          A       B
1       NAME    SCORE
2       Anna     65.8
3       Bill     95.4
4       Donna    30.2
5       Mark     54.9
6       Maria    35.1
7       Susan    75.9
8       John     83.2
9       Rob      33.1
10      Ethan    81.8

@STD(B2:B10) -> 23.1
```

Note: STDDEV() and STDEV() differ from @STD() in that STDDEV() and STDEV() ignore string values while @STD() treats string values as zero.

SUM or @SUM

Summary Adds all the numbers in a list.

Usage SUM(*num,...*), SUM(*range*), SUM(*range1, range2,...*)

Arguments

num any number
range any worksheet range

Description This function finds the sum of all the numbers in the given list or range. The items in the list are separated by commas, and they may be numbers or cell or range references containing numbers. You can specify as many numbers or ranges as you want. You may list any combination of numbers, cell addresses, and ranges as arguments.

Examples

```
=SUM(300,400,2300,100) -> 3100

A1:  Blue
A2:  300
A4:  900
B1:  1200
=SUM(A1:A4,B1) -> 2400
```

TAN or @TAN

Summary Returns the tangent of the given number .

Usage TAN(*num*)

Arguments

num any number

Description The tangent of an angle is ratio of the Y coordinate to the X coordinate of the point where a line at that angle from the X axis intersects a circle of radius one (1) centered on the origin. The TAN function takes the value of the angle in radians; the value of the angle can be converted from degrees to radians

within this function by using it in conjunction with the DEGTORAD function. See "DEGTORAD or @DEGTORAD" on page 206.

Examples

```
=TAN(1.24) -> 2.9119
```

VAR

Summary

Calculates the statistical variance of a list of numbers. Strings and blank cells are not counted.

Usage

VAR(*num,...*), VAR(*range*), VAR(*range1, range2,...*)

Arguments

num any number
range any worksheet range

Description

This function finds the statistical variance of a range of list of numbers. The items in the list are separated by commas, and they may be numbers or cell/range references containing numbers. You may list any combination of numbers, cell addresses, and ranges as arguments. Variance is a measure of how much a set of numbers vary from each other. Strings and blank cells in the list are ignored.

Examples

```
          A       B
1       NAME    SCORE
2       Anna     65.8
3       Bill     95.4
4       Donna    30.2
5       Mark     54.9
6       Marie    35.1
7       Susan    75.9
8       John     83.2
9       Rob      33.1
10      Ethan    81.8

=VAR(B2:B10) -> 531.6010
```

@VAR

Summary

Calculates the statistical variance of a list of numbers. Strings are counted as 0, and blank cells are not counted.

Usage

@VAR(*num,...*), @VAR(*range*), @VAR(*range1, range2,...*)

Arguments

num any number
range any worksheet range

Description

This function finds the statistical variance of a range of list of numbers. The items in the list are separated by commas, and they may be numbers or cell/range references containing numbers. You may list any combination of numbers, cell addresses, and ranges as arguments. Variance is a measure of how much a set of numbers vary from each other. This function is essentially the same as VAR, except that it treats strings as having a value of zero.

Examples

```
          A       B
1       NAME    SCORE
2       Anna     65.8
3       Bill     95.4
4       Donna    30.2
5       Mark     54.9
6       Marie    35.1
7       Susan    75.9
8       John     83.2
9       Rob      33.1
10      Ethan    81.8

@VAR(B2:B10) = 531.6010
```

Note: VAR and @VAR are different in that VAR ignores strings while @VAR treats string values as zero.

Financial Functions

CALCRATE or @CALCRATE

Summary

Calculates the interest rate on an investment, given its present value, the amount of each payment, and the number of payment periods.

Usage

CALCRATE(*pv*, *pmt* , *per*)

Arguments

pv	the present value of the investment
pmt	the amount of each payment
per	the number of payment periods

Description

Given an investment with a present value of *pv*, payments of *pmt* , and *per* payment periods, the CALCRATE function returns the interest rate such that *pv* = @PV(*pmt* , *rate* , *term*). See "@PV" on page 235.

Examples

```
Find the monthly interest rate of a 36 month car
loan on $17,000 with a monthly payment of $500:
=CALCRATE(17000,500,36) = 0.31% per month or 3.75%
per year.
```

@CTERM

Summary

Computes the number of periods required for an investment to grow to the specified Future V alue with a given interest rate.

Usage

@CTERM(*rate* , *fv*, *pv*)

Arguments

rate	the interest rate
fv	the desired Future V alue of the investment
pv	the Present V alue of the investment

Description

Given an investment with a Present V alue of *pv*, the @CTERM function returns the number of periods required for it to grow to a Future V alue of *fv* at an interest rate of *rate* .

Examples

```
How long will it take for a $20,000 investment to
grow to $40,000 at an annual rate of 5%?
@CTERM(0.05,20000,40000) -> 14.2 years

How many months will it take for a $4,000 invest-
ment to grow to $8,000, being compounded monthly
at a rate of 8%?
@CTERM(8%/12,4000,8000) -> 104 months (about 8 1/2
years)
```

DDB or **@DDB**

Summary

Computes the Double Declining Balance depreciation for an asset.

Usage

DDB(*cost* , *salvage* , *life* , *per*)

Arguments

cost	the original cost of the asset
salvage	the salvage value of the asset
life	the expected life of the asset
per	the number or periods for the depreciation calculation

Description

Given an asset that has a cost of *cost* , a salvage value of *salvage* , and a life of *life* periods, the DDB function will return the depreciated value of the asset after *per* periods.

Examples

```
A company buys a delivery van for $12,000. It has a
useful life of 5 years and will be worth $2,000 at
the end of that time. How much depreciation can
the company write off in years 1,2 and 3?
       year 1:  =DDB(12000,2000,5,1) -> $4,800
       year 2:  =DDB(12000,2000,5,2) -> $2,880
       year 3:  =DDB(12000,2000,5,3) -> $1,728
```

FV or @FV

Summary

Computes the Future V alue of a series of payments, given a specified interest rate. An optional present value and starting time for the first payment may be specified.

Usage

FV(*rate* , *per*, *pmt* [, *pv* [, *type*]])

Arguments

rate	the interest rate
per	the number of payments
pmt	the size of each payment
pv	the Present V alue
type	the starting time of the first payment

Description

Returns the Future V alue of a stream of payments over *per* periods at *rate* interest rate. If present value *pv* is not specified, it is assumed to be 0. *Type* designates when the payments start.

Examples

```
If $2,000 were deposited every year into an
account earning 9% annual interest, how much would
the account be worth in 20 years?
       =FV(9%,20,-2000) -> $102,320.24
If a $50,000 were deposited instead and accrued 9%
interest from the start of the year, what would
the account be worth in 20 years?
       =FV(9%,20,,-50000,1) -> $ 280,220.54
```

IRR

Summary

Calculates the Internal Rate of Return of an investment.

Usage

IRR(*guess* , *num* 1, *num2, ...*), IRR(*guess, range1, ..* .)

Arguments

guess
num
range

Description

IRR() calculates the internal rate of return for a series of cash flow values. Each value represents either a payout (a negative value) or a payment received (positive value). The internal rate of return is the interest rate used to discount future cash flows such that the net value of all the cash flows is zero. You may list any combination of numbers, cell addresses, and ranges as arguments.

Examples

```
A rug-cleaning business is being offered for sale
for $180,000.  The owner says the business should
net $25,000, $35,000, $39,000, $40,000 and $55,000
in years 1-5.  What is this project's internal
rate of return?
         A
1     -180000
2       25000
3       35000
4       39000
5       40000
6       55000

The initial expense is entered as a negative num-
ber in cell A1.  The range of cash inflows are
given in cells A2:A6. To calculate this project's
IRR value:

=IRR(B1,A1:A6) -> 2.3%
```

@IRR

Summary

Computes the Internal Rate of Return of an investment. String entries are considered zero values.

Usage

@IRR(*guess* , *num* 1, *num2, ...*), @IRR(*guess, range1, ..* .)

Arguments

guess
num
range

Description

@IRR() calculates the internal rate of return for a series of cash flow values. Each value represents either a payout (a negative value) or a payment received (positive value). The internal rate of return is the interest rate used to discount future cash flows such that the net value of all the cash flows is zero. You may list any combination of numbers, cell addresses, and ranges as arguments.

Examples

```
A rug-cleaning business is being offered for sale
for $180,000. The owner says the business should
net $25,000, $35,000, $39,000, $40,000 and $55,000
in years 1-5. What is this project's internal rate
of return?
         A
1     -180000
2       25000
3       35000
4       39000
5       40000
6       55000

The initial expense is entered as a negative num-
ber in cell A1. The range of cash inflows are given
in cells A2:A6. To calculate this project's IRR
value:

@IRR(B1,A1:A6) -> 2.3%
```

NPV

Summary

Computes the Net Present V alue of an investment with a given stream of income at a given interest rate. String values are ignored.

Usage

NPV(*rate* , *pmt1* [, *pmt2* [, ...]])

Arguments

rate the interest rate
pmt a number or range — the payment amounts

Description

Given an interest rate of *rate* and payments *pmt1* , *pmt2* , etc., the NPV function returns the Net Present Value of the stream of payments. Each payment can be a value or a range. If it is a range, each of the numbers in the range is included in the calculation. You may list any combination of numbers, cell addresses, and ranges as arguments for *pmt* .

Examples

```
An apartment building is being offered for sale
for $400,000. Expected rental incomes for years 1-
5 are $79,000, $85,000, $85,000, $103,000 and
$120,000 and the interest rate is expected to stay
close to 7%. Is this a worthwhile investment?

          A
1      -400000
2        79000
3        85000
4        85000
5       103000
6       120000

To figure out the net present value of the cash in-
flows:
=NPV(7%,A2:A6) -> $381,596

To figure out the value of the project:
     Project Value = Initial Cost + NPV
          +A1+NPV(7%,A2:A6) -> -$18,404
No, this project is not worthwhile.
```

@NPV

Summary

Computes the Net Present V alue of an investment with a given stream of income at a given interest rate. String values are treated as zeros.

Usage

NPV(*rate* , *pmt1* [, *pmt2* [, ...]])

Arguments

rate the interest rate
pmt a number or a range — the payment amounts

Description

Given an interest rate of *rate* and payments *pmt1* , *pmt2* , etc., the NPV function returns the Net Present Value of the stream of payments. Each payment can be a value or a range. If it is a range, each of the numbers in the range is included in the calculation. You may list any combination of numbers, cell addresses, and ranges as arguments for *pmt* .

Examples

```
An apartment building is being offered for sale
for $400,000. Expected rental incomes for years 1-
5 are $79,000, $85,000, $85,000, $103,000 and
$120,000 and the interest rate is expected to stay
close to 7%. Is this a worthwhile investment?

          A
1      -400000
2        79000
3        85000
4        85000
5       103000
6       120000

To figure out the net present value of the cash in-
flows:
@NPV(7%,A2:A6) -> $381,595

To figure out the value of the project:
     Project Value = Initial Cost + NPV
          +A1+@NPV(7%,A2:A6) -> -18,404
No, this project is not worthwhile.
```

PMT

Summary

Calculates the size of payments required to pay off a loan.

Usage

PMT(*rate* , *per*, *loan* [, *fv* [, *type*]])

Arguments

loan	the amount borrowed
rate	the interest rate
per	the number of payment periods
fv	the [optional] Future V alue of the loan
type	the [optional starting time of the first payment

Description

Given a loan of *loan* , an interest rate of *rate* , and *per* payment periods, the @PMT function returns the size of the payments. If the Future V alue *fv* is not specified, it is assumed to be 0. If *type* is not specified, payments are assumed to start immediately .

Examples

```
What would be the monthly payment on an $40,000
home improvement loan, to be repaid in 5 years at a
fixed rate of 9%?
 =PMT(9%/12,5*12,-40000) -> $830.33 per month.
```

@PMT

Summary Calculates the size of payments required to pay off a loan.

Usage @PMT(*prin* , *rate* , *per*)

Arguments

loan	the amount borrowed
rate	the interest rate
per	the number of payment periods

Description Given a loan of *loan* , an interest rate of *rate* , and *per* payment periods, the @PMT function returns the size of the payments required.

Examples

```
What would be the monthly payment on an $40,000
home improvement loan, to be repaid in 5 years at a
fixed rate of 9%?

@PMT(40000,9%/12,5*12) -> $830.33 per month.
```

PV

Summary Computes the Present V alue of an investment.

Usage PV(*rate* , *per*, *pmt* [, *fv* [, *type*]])

Arguments

rate	the interest rate
per	the number of payment periods
pmt	the amount of each payment
fv	the [optional] Future V alue of the investment
type	the [optional] starting time of the payments

Description Given an interest rate of *rate* , *per* payment periods, and a periodic payment of *pmt* , the PV function returns the Present V alue of the investment. If a

future value *fv* is not given, it is assumed to be 0. If *type* is not zero, payments are assumed to start in the next period.

Examples

```
What is the present value of the retirement annu-
ity which promises to pay $30,000 every year for
the next 20 years, given an interest rate of 12%?

=PV(12%,20,-30000) -> $224,083.31

What is the present value of the investment which
promises to pay $225,000 at the end of 20 years,
given an interest rate of 12%?

=PV(12%,20,-225000) -> $23,325.02
```

@PV

Summary Computes the Present V alue of an investment.

Usage @PV(*pmt* , *rate* , *per*)

Arguments

pmt the amount of each payment
rate the interest rate
per the number of payment periods

Description Given *per* payments of size *pmt* and an interest rate of *rate* , the @PV function returns the Present V alue of the investment.

Examples

```
What is the present value of the retirement annu-
ity which promises to pay $30,000 every year for
the next 20 years, given an interest rate of 12%?

@PV(30000,12%,20) -> $224,083.31
```

@RATE

Summary

Returns the interest rate for an investment, given the Future V alue, Present V alue, and number of payment periods.

Usage

@RATE(*fv*, *pv*, *per*)

Arguments

fv	the desired Future V alue of the investment
pv	the Present V alue of the investment
per	the number of payment periods

Description

Given a Present V alue of *pv*, the @RA TE function returns the required interest rate necessary to achieve the desired Future V alue of *fv* in *per* payment periods.

Examples

```
How much interest would a bank account need to
earn for an investment of $67,000 to grow to
$100,000 in 5 years or less?

@RATE(100000,67000,5) -> 8.34% or more.
```

SLN or @SLN

Summary Returns the Straight-Line depreciation of an asset.

Usage SLN(*cost* , *slvg* , *life*)

Arguments

cost	the original cost of the asset
slvg	the salvage value of the asset
life	the expected lifetime of the asset

Description Given an asset with an original cost of *cost* , a salvage value of *slvg* , and a lifetime of *life* , the SLN function returns the Straight-Line depreciation for the asset.

Examples

```
A company buys a delivery van for $12,000. It has a
useful life of 5 years and will be worth $2,000 at
the end of that time. How much depreciation can
the company write off each year?

=SLN(12000,2000,5) -> $2,000
```

SYD or @SYD

Summary

Calculates the Sum of the Y ears Digits depreciation for an asset.

Usage

SYD(*cost* , *slvg* , *life* , *per*)

Arguments

cost	the original cost of the asset
slvg	the salvage value of the asset
life	the expected lifetime of the asset
per	the number of periods for the depreciation calculation

Description

Given an asset with an original cost of *cost*, a salvage value of slvg, and a lifetime of *life* periods, the SYD function returns the Sum of the Y ears Digits depreciation for *per* periods.

Examples

```
A company buys a delivery van for $12,000. It has a
useful life of 5 years and will be worth $2,000 at
the end of that time. How much depreciation can
the company write off in years 1,2 and 4?

year 1:  =SYD(12000,2000,5,1) -> $3,333
year 2:  =SYD(12000,2000,5,2) -> $2,667
year 4:  =SYD(12000,2000,5,4) -> $1,333
```

@TERM

Summary

Computes the number of payments required to reach a desired Future V alue with a given payment size and interest rate.

Usage

@TERM(*pmt* ,*rate* ,*fv*)

Arguments

pmt	the size of each payment
rate	the interest rate
fv	the desired Future V alue

Description

Given payments of size *pmt* and an interest rate of *rate* , the @TERM function returns the number of payments required to reach the desired Future V alue *fv*.

Examples

```
How many $700 monthly payments would be needed to
accumulate $250,000 in a bank account earning 4%
interest?

@TERM(700,4%/12,250000) -> 235.6 monthly payments
(taking almost 20 years).
```

String Functions

CHAR or @CHAR

Summary Returns the ASCII character for the given number .

Usage CHAR(*num*)

Arguments *num* any integer representing an ASCII character

Description The CHAR function will convert the integer *num* to its corresponding ASCII character . The ASCII code for a given character can be found with the CODE function. See “CODE or @CODE” on page 241.

Examples

```
=CHAR(68) -> D
@CHAR(162) -> ¢
```

CLEAN or @CLEAN

Summary Removes all control characters from a string.

Usage CLEAN(*string*)

Arguments *string* any string

Description The CLEAN function will remove all control characters from the string *string* . This can be useful for data that has been imported to Mesa from another source.

CODE or @CODE

Summary Returns the ASCII code for the first character in a string.

Usage CODE(*string*)

Arguments

string any string (only the first character will be used)

Description The CODE function returns the ASCII code for the first character in the string *string* . The character for a given code can be found with the CHAR function. See "CHAR or @CHAR" on page 240.

Examples

```
=CODE("A") -> 65
@CODE("{") -> 123
=CODE("Hello") -> 72  :  This formula is returning
the ASCII code for "H"
```

EXACT or @EXACT

Summary Compares two strings to determine if they are exact matches.

Usage EXACT(*str1* , *str2*)

Arguments

str1 the first string to be compared
str2 the second string to be compared

Description The EXACT function compares *str1* and *str2* , and returns 1 if the strings match exactly . It returns 0 otherwise. The EXACT function is case sensitive. When you use = to match strings, the comparison ignores case. For example, "Upper" = "upper" is true, but EXACT("Upper", "upper") is false.

Examples

```
=EXACT("Daytime","Daytyme") -> 0 (false)
A1="hello" and B1="hello": =EXACT(A1,B1) -> 1
(true)
```

FIND or @FIND

Summary Finds the location of a substring within a string.

Usage FIND(*substr* , *string* , *offset*)

Arguments

substr the substring to be found
string the string to be searched
offset the location in the string to begin the search

Description The FIND function returns the location of the sub-string *substr* in the string *string* . It starts searching for *substr* *offset* characters from the left of *string* . It returns ERR if the string is not found.

Examples

```
=FIND("hampton","Northampton",1) -> 4

=FIND("hampton","Northampton",6) -> !ERR (because
it starts searching at the 6th character)

A1:  "The third word in this sentence."
=FIND("word",A1,1) -> 10
```

FIXED or @STRING

Summary

Rounds a number to a specified number of decimal places and converts it to a string.

Usage

FIXED(*num* , *prec*)

Arguments

num any number
prec the number of places of precision

Description

The FIXED function rounds the number *num* to a precision of *prec* decimal places, then converts it to a string. If *prec* is negative, *num* will be rounded off to the left of the decimal point.

Examples

```
B3:  5251.769585987
=FIXED(B3,6) = 5251.769586

=FIXED(3/43,2)&" is now  a string" -> 0.07 is now a
string  :  This formula evaluates 3/43, then con-
verts it to a string with 2 decimal places of pre-
cision and concatenates it with the text " is now a
string".
```

LEFT or @LEFT

Summary

Returns a substring from the left of a string.

Usage

LEFT(*string* , *len*)

Arguments

string any string
len the length of the substring

Description

The LEFT function returns a string containing the first *len* characters from the left of the string *string* .

Examples

```
=LEFT("The first twelve characters",12) -> The
first tw
=LEFT("ABCDEFG",4) -> ABCD
```

LEN, LENGTH, @LENGTH or @LEN

Summary Returns the length of a string.

Usage LENGTH(*string*)

Arguments *string* any string

Description The LENGTH function returns the number of characters in the string *string* .

Examples

```
LENGTH("10% discount on COD") = 19
A1:  "A short string "
B1:  "sentence"
=LEN(A1) -> 15
=LENGTH(A1&B1) -> 23
```

LOWER or @LOWER

Summary Converts a string to lower case.

Usage LOWER(*string*)

Arguments *string* any string

Description The LOWER function converts all the characters in the string *string* to lower case, regardless of their original case.

Examples

```
=LOWER("Bill Johnson") -> bill johnson
```

MID or @MID

Summary Returns a substring from the middle of a string.

Usage MID(*string* , *start* , *len*)

Arguments

string	any string
start	the starting character of the substring
len	the length of the substring

Summary The MID function returns a string of *len* characters from the string *string* , starting *start* characters from the first character of *string* .

Examples

```
= MID("Have a great day!",7,5) = great

A1:  "Robert Quimby Douglas",
=MID(A1,7,6) -> Quimby
```

NEXT or @NEXT

Summary Returns the next logical value for a given value

Usage NEXT(*value* [,*inc*])

Arguments

value	a number , string, or cell reference
inc	the [optional] number by which to increment the value

Description The NEXT function returns the next logical value after *value* . The value is incremented by 1 unless an optional increment *inc* is specified. For example, if *value* is “Mon”, then NEXT(*value*) returns “T ue”. Values for months, days, and quarters are built in. If *value* is a number , NEXT adds 1 or *inc* to the number . For quarterly information, it will increment the

quarter . When the quarter reaches 4, the next quarter is Q1 of the following year . For example, NEXT("Q4 '92") is "Q1 '93".

Examples

```
=NEXT("Q4 '92") -> Q1 '93
=NEXT("January") -> February
=NEXT("Region 1") -> Region 2
```

PROPER or @PROPER

Summary Converts a string to Proper Case.

Usage PROPER(*string*)

Arguments *string* any string

Description The PROPER function capitalizes the first letter of every word in the string *string* and converts all remaining characters to lower case.

Examples

```
=PROPER("MS. JANET KRAFT") -> Ms. Janet Kraft
=PROPER("all entries posted before 7/2") -> All
Entries Posted Before 7/2
```

Note: A bit of history: Upper and lower case does not refer to the size of the letters, but to where they were stored in type cabinets. The capital letters were stored in the "Upper Case," while the small letters were stored in the "Lower Case."

REPEAT, REPT, or @REPEAT

Summary Repeats a string the specified number of times.

Usage REPEA T(*string* , *num*)

Arguments

string any string
num the number of times to repeat the string

Description This function returns the string *string* repeated *num* times. The string is placed back-to-back with itself, with no white space.

Examples

```
=REPEAT("X",10) -> XXXXXXXXXX
=REPT("over and ",4) -> over and over and over and
over and
```

REPLACE or @REPLACE

Summary Replaces a specified number of characters in a string with another string.

Usage REPLACE(*string* , *offset* , *num* , *newstr*)

Arguments

string any string
offset the position to begin the insertion
num the number of characters to replace
newstr the string with which to replace the specified characters

Description The REPLACE() function replaces *num* characters in the string *string* with the string *newstr* , starting *offset* characters from the first character of the original

string. The string *newstr* does not need to be the same length as the number of characters being replaced.

Examples

```
=REPLACE("The cost is $450, COD",13,3,"2,500") ->
The cost is $2,500, COD
=REPLACE("Robert J. Stevenson",7,1,"L") -> Robert
L. Stevenson
```

RIGHT or @RIGHT

Summary Returns a substring from the right of a string.

Usage RIGHT(*string* , *len*)

Arguments

string any string
len the length of the substring

Description The RIGHT function returns a string containing the first *len* characters from the right of the string *string* .

Examples

```
=RIGHT("The last ten characters",10) -> characters
=RIGHT("Ida M. Brown",5) -> Brown
```

TRIM or @TRIM

Summary Removes extra white space from a string

Usage TRIM(*string*)

Arguments *string* any string

Description This function removes all leading and trailing spaces from the string *string* and converts multiple consecutive spaces within the string to single spaces.

Examples

```
A3:  Account #        3040302     40303
=TRIM(A3) -> Account # 3040302 40303

=TRIM("    Anything    at    all.") -> Anything at
all.
```

UPPER or @UPPER

Summary Converts a string to UPPER case.

Usage UPPER(*string*)

Arguments *string* any string

Description The UPPER function converts all the characters in the string *string* to upper case, regardless of their original case.

Examples

```
=UPPER("Uppers only") -> UPPERS ONLY

=UPPER(MID("robert c. smith",8,2)) -> C.  :  This
formula uses MID() to return the 8th and 9th char-
acters from the string "Robert C. Smith".  The
UPPER() function then capitalizes this result.
```

VALUE

Summary Returns the number value of a string.

Usage VALUE(*string*)

Arguments *string* any string

Description This function converts the string *string* into a number. If the string begins with a number but contains non-numbers, VALUE() returns as much as it can. If *string* begins with a letter, it returns 0.

Examples

```
=VALUE("3.14159") = 3.14159
=VALUE("1200") + 2000 = 3200
=VALUE("320/104") = 320
```

@VALUE

Summary Returns the number value of a string.

Usage @VALUE(*string*)

Arguments *string* any string

Description This function converts the string *string* into a number. If the string contains characters that are non-numbers, it returns an error .

Examples

```
@VALUE("3.14159") -> 3.14159
@VALUE("1200") + 2000 -> 3200
@VALUE("320/104") -> !ERR
```

Note: VALUE() and @VALUE() are different in that if the argument to VALUE() cannot be completely converted to a number, it returns as much of the number as possible. @VALUE() returns an error .

Special Functions

#NOT#

Summary Logical NOT .

Usage #NOT# *arg*

Arguments *arg* a number or true/false value

Description This is the logical NOT function, but it is not used the same way as the logical AND and OR. For example, *#NOT# 1* will equal 0, while *#NOT# 0* will equal 1.

Examples

```
A3 = +("Blue"="Red") ,  which returns 0 #NOT#A3   =
1   :  The formula in cell A3 asks "is the string
'Blue' = the string 'Red'?  The answer is 0
(false).  The #NOT# function inverts this return-
ing the answer 1 (true)

#NOT#1 = 0 (false)
```

@@

Summary Evaluates a string as a cell address.

Usage @@(*string*)

Arguments

string any string that can be interpreted as a cell address

Description This function evaluates the string *string* as a cell address and returns that address. This can be useful for creating a cell address with string concatenation.

Examples

```
A3: Cheers
B3: A3
@@(B3) -> Cheers

A1: 10
A2: 20
A3: 30
B1: A1:A3
@SUM(@@(B1)) -> 60
```

ADDRESS or @ADDRESS

Summary Returns a cell address for a given row and column number .

Usage ADDRESS(*row*,*col*)

Arguments

row the number of the row
col the number of the column

Description The ADDRESS() function returns the address for the *row* and *col* specified. Row and column numbering starts at zero, i.e. ADDRESS(0,0) is cell A1. This function can be useful in MScript scripts.

Examples

```
B1: 15
=ADDRESS(0,1) -> 15
```

BEEPIF or @BEEPIF

Summary Beeps if the specified condition is true.

Usage BEEPIF(*cond*)

Arguments *cond* a conditional expression

Description The BEEPIF function beeps and returns 1 if the condition *cond* is true, and returns 0 if the condition if false. This can be useful for alerting the user of certain conditions within the worksheet.

Examples

```
C7:  100
=BEEPIF(C7<200) :  This formula will beep one
time.

=BEEPIF(rand>0.5) : This formula will generate a
random number between 0 and 1.  If the number is >
0.5, the it beeps and returns 1.  Otherwise it
returns 0.
```

CELL or @CELL

Summary

Returns information about the attributes of a given cell.

Usage

CELL(*atrb* ,*cell*)

Arguments

atrb the cell attribute
cell the cell address

Description

The CELL() function returns attribute information about a the cell at the address *cell*. The type of information requested is specified by the argument *atrb* , chosen from the list below .

Examples

```
U7: PI

=CELL("address",U7) -> $U$7
=CELL("row",U7) -> 7
=CELL("col",U7) -> 21
=CELL("contents",U7) -> 3.141593
```

ADDRESS - returns the address of the cell
ROW - returns the row of the cell
COL - returns the column of the cell
CONTENTS - returns the contents of the cell
FORMAT - returns the format of the cell. Formats are:

- G - General
- Fn - Fixed Decimal n places
- Sn - Scientific Format n decimal places
- Cn - Currency format n decimal places
- ,n - Comma format n decimal places
- + - Chart Format
- Pn - Percent Format n decimal places
- T - Text Format
- H - Hidden Format
- D1 - Day Month Year Format
- D2 - Day Month Format
- D3 - Month Year Format

D6 - Hours Minutes Seconds Format
D7 - Hours Minutes Format
D4 - International Date Format 1
D5 - International Date Format 2
D8 - International Time Format 1
D9 - International Time Format 2

PREFIX - The string prefix in Lotus Format
PROTECT - Returns 0 if the cell is Unprotected, 1 otherwise
TYPE - b - blank cell, v - number value, l - string value
WIDTH - Returns the width of the column in characters
RANGE - Converts a single address to a range

CELLPOINTER or @CELLPOINTER

Summary Returns information about the attributes of the currently active cell.

Usage CELLPOINTER(*atrb*)

Arguments *atrb* the cell attribute

Description The CELLPOINTER function returns attribute information about a the currently active cell. The type of information requested is specified by the argument *atrb* . See “CELL or @CELL” on page 254 for information about attributes. The currently active cell is the cell on the worksheet whose contents is currently being displayed in the formula bar . Mesa worksheets always have one currently active cell. If a range has been selected, the currently active cell is the cell in the upper left corner of the range.

CHOOSE or @CHOOSE

Summary Chooses a numbered item from a list.

Usage CHOOSE(*num* , *item1* [, *item2* [, ...])

Arguments

num the number of the chosen item
item the members of the list

Description Chooses the *num* item from the list of items *item1* , *item2* , etc. This function counts the listed items starting at zero. You may list any combination of numbers, cell addresses, and ranges as arguments for *item* .

Examples

```
=CHOOSE(2,"first","second","third","fourth") ->
third

=CHOOSE(MOD(INT(NOW),7),"Thu","Fri","Sat",Sun","Mo
n","Tue","Wed") : This function returns the cur-
rent day using NOW to provide the current date.
```

COLS or @COLS

Summary Returns the number of columns in a range.

Usage COLS(*range*)

Arguments

range any range

Description The COLS function returns the number of columns contained in the range *range* .

Examples

```
=COLS(A1:H20) -> 8

If RANGE1 is the name assigned to the range
B1:CC200, =COLS(RANGE1) -> 80
```

DIDCHANGE or @DIDCHANGE

Summary Determines if the sheet has changed since it was last saved.

Usage DIDCHANGE

Arguments <none>

Description DIDCHANGE returns a 1 if the sheet has changed since the last time it was saved, and a 0 if nothing has changed. Since the act of adding this function changes the sheet, it will only show as true when the sheet has been recalculated right after it has been saved. This can be useful in MScript scripts.

FORMAT or @FORMAT

Summary Formats the number in the predefined format

Usage FORMA T(*num* , *fmt-const* [,*prec*])

Arguments

num	any number
fmt-const	any format constant
prec	any integer [optional]

Description This function returns the given number *num* in the format specified by the format constant *fmt-const* . A list of the format constants can be found in “Format Constants” on page 301. You may optionally specify the precision *prec*, the number of decimal places in the number .

Examples

```
=FORMAT(18.9, SCIENTIFICFORMAT) -> 1.89e+01
```

GETINPUT or @GETINPUT

Summary Returns the user -typed input string.

Usage GETINPUT(*cell*)

Arguments *cell* any cell

Description The GETINPUT function returns the string typed into the cell *cell* by the user , rather than its value. This can be useful for MScript scripts.

Examples

```
A1 : =INT(3.2)
=A1 is 3
=GETINPUT(A1) is the string =INT(3.2)
```

HISTOGRAM or @HISTOGRAM

Summary Creates a histogram for a given range.

Usage HISTOGRAM(*range* , *divs* , *anchor* [, *min* , *max*])

Arguments

range any range
divs the number of divisions in the histogram
anchor the starting cell of the histogram
min the [optional] minimum
max the [optional] maximum

Description The HISTOGRAM function groups the values in *range* into *divs* divisions and counts the number of items in each division. The formula should appear in *divs* cells, starting at *anchor* . It returns the number of items in the division that is offset from the anchor point. *Min* and *max* are optional parameters that set

the minimum and maximum values of the histogram range, so that the distribution will only be calculated for values between *min* and *max*.

Examples

```
     A      B
1  Name    GPA
2  Joe     2.6
3  Sue     3.7
4  Robert  3.2
5  Steve   3.0
6  Ann     3.8
7  John    3.5
8  Mary    2.9
9  Chris   3.5

To divide the students into 5 divisions by GPA and
determine how many fall into each division:
A10: =HISTOGRAM($B$2:$B$9,5,$A$10) -> 1
B10: =SAME($A$10) -> 2
C10: =SAME($A$10) -> 1
D10: =SAME($A$10) -> 2
E10: =SAME($A$10) -> 2
```

HLOOKUP or @HLOOKUP

Summary

Looks up a code in a horizontal table.

Usage

HLOOKUP(*code*, *table*, *offset*)

Arguments

code	the code to be looked up
table	a range containing the lookup data
offset	the number of columns offset from the top of the table.

Description

Returns the cell *offset* cells from the top of the first match. If the first row contains numbers and *code* is less than the first value in the lookup row, !ERR is returned. If *code* is greater than the first value but no

exact match is found, HLOOKUP returns the value from the last column. If the first row contains only strings and no exact match is found, it returns !ERR.

Examples

```
     A    B    C
1     1    2    3
2  $200  $40 $500

=HLOOKUP(2,A1:C2,1) -> $400
=HLOOKUP(0.5,A1:C2,1) -> !ERR
=HLOOKUP(5,A1:C2,1) -> $500
```

IF or @IF

Summary

Returns a conditional result.

Usage

IF(*exp* ,*value1* [,*value2*])

Arguments

exp the conditional expression for evaluation
value1 the first possible result
value2 the [optional] second possible result

Description

The IF function evaluates *exp* . If it is a true or non-zero value, the result is *value1* . If it is false, zero, or a string, optional *value2* or an empty value is the result.

Examples

```
=IF(5>10,"Something is wrong!","Everything is nor-
mal.")
-> Everything is normal.

A1: 3400
A2: 4000
A3: 5000
=IF(SUM(A1:A3)>=12400,"Full Capacity") -> Full
Capacity
```

IFELSE

Summary Returns a conditional result.

Usage IFELSE(*exp*, *value1* , *value2*)

Arguments

exp the conditional expression for evaluation
value1 the first possible result
value2 the second possible result

Description The IFELSE() function evaluates *exp*. If it is a true or non-zero value, the result is *value1* . If it is false, zero, or a string, *value2* is the result.

Examples

```
=IFELSE(5>10,"Something is wrong!","Everything is
normal.") -> Everything is normal.

A1: 3400
A2: 4000
A3: 5000
=IFELSE(SUM(A1:A3)>40000,"Full Capacity",”Okay”) -
> Okay
```

INDEX

Summary Returns a specified cell from a range of cells.

Usage INDEX(*range* , *row*, *col*)

Arguments

range	any range
row	the row offset from the upper left of the range
col	the column offset from the upper left of the range

Description The INDEX function returns the cell that is *col* columns and *row* rows offset from the upper left corner of the range *range* .

Examples

```
     A     B     C
1     10    20    30
2   3942  4932  5929
3   4920  5929  9294
=INDEX(A1:C3,2,2) -> 9294

A10: 1
A11: 0
=INDEX(A1:C3,A10,A11) -> 3942
```

@INDEX

Summary Returns a given cell from a range of cells.

Usage INDEX(*range* ,*col*,*row*)

Arguments

range any range
col the column offset from the upper left of the range
row the row offset from the upper left of the range

Description The INDEX() function returns the cell that is *row* rows and *col* columns offset from the upper left corner of the range *range* .

Examples

```
     A     B     C
1     10    20    30
2   3942  4932  5929
3   4920  5929  9294
@INDEX(A1:C3,2,2) = 9294

A10: 0
A11: 1
@INDEX(A1:C3,A10,A11) = 3942
```

Note: INDEX and @INDEX differ slightly . INDEX() specifies the offset in row then column order . @INDEX() specifies the offset in column then row order .

ISEMPTY or @ISEMPTY

Summary Determines if the cell at the given address is empty .

Usage ISEMPTY(*adr*)

Arguments *adr* a cell address

Description This functions tests whether or not the specified cell is empty . Mesa functions that return a T rue or False result return 1 for T rue, and 0 for False.

Examples

```
A1 : 5.2
B1 :
=ISEMPTY(A1) is 0
=ISEMPTY(B1) is 1 because cell B1 has no contents.
```

ISERROR or @ISERR

Summary Determines if the given expression returns an error other than @NA.

Usage ISERROR(*exp*)

Arguments *exp* an expression

Description This function tests whether or not the given expression results in an error . In Mesa functions that return a T rue/False result, 1 corresponds to T rue, and 0 corresponds to False. See “ISNA or @ISNA” on page 265.

Examples

```
=ISERR(ERRORCONST) -> 1 (true)

=ISERR(@VALUE("ABCD") -> 1 (true)  :  This formula
tried to convert "ABCD" to a value using @VALUE().
This returns !ERR, which causes ISERR() to returns
1.
```

ISFORMULA or @ISFORMULA

Summary Determines if the cell at the given address contains a formula.

Usage ISFORMULA(*adr*)

Arguments *adr* a cell address

Description This function tests to see if the cell at the given address contains a formula. In Mesa functions that return a T rue/False result, 1 corresponds to T rue, and 0 corresponds to False.

Examples

```
A1 : 3.2
B1 : =A1+1
=ISFORMULA(A1) is 0
@ISFORMULA(B1) is 1 because cell B1 contains a
formula that adds 1 to the contents of cell A1.
```

ISNA or @ISNA

Summary Determines if the given expression returns an NA error .

Usage ISNA(*exp*)

Arguments *exp* an expression

Description This function tests to see if the given expression returns the NA error . In Mesa functions that return a True/False result, 1 corresponds to T rue, and 0 corresponds to False.

Examples

```
=ISNA(@NA) -> 1 (true)

A1: 34
A2: 49
A3: =NA
=ISNA(SUM(A1:A3)) -> 1  :  This formula returns 1
(true) because cell A3 contains NA, which causes
SUM(A1:A3) to evaluate to NA also.
```

ISNUMBER or @ISNUMBER

Summary Determines if the given expression returns a numeric result.

Summary ISNUMBER(*exp*)

Arguments *exp* an expression

Description This function tests to see whether or not the given expression results in a number . In Mesa functions that return a T rue/False result, 1 corresponds to True, and 0 corresponds to False.

Examples

```
=ISNUMBER(FIXED(300,0)) -> 0  :  This formula con-
verts the number "300" to a string using FIXED().
It then uses ISNUMBER() to test if that result is a
value and returns 0 (false).

=ISNUMBER(34) -> 1

A3: +394 - 100
=ISNUMBER(A3) -> 1
```

ISREF or @ISREF

Summary Determines whether or not the given expression is a cell or range reference.

Usage ISREF(*exp*)

Arguments *exp* an expression

Description This function tests to see whether or not the given expression is a cell or range reference. In Mesa functions that return a T rue/False result, 1 corresponds to T rue, and 0 corresponds to False.

Examples

```
=ISREF(A1) -> 1 (true)

If RANGE1 has been assigned A1:C10, =ISREF(RANGE1)
-> 1
```

ISSTRING or @ISSTRING

Summary Determines whether or not the given expression results in a string.

Usage ISSTRING(*exp*)

Arguments *exp* an expression

Description This function tests whether or not the given expression will result in a string when evaluated. In Mesa functions that return a T rue/False result, 1 corresponds to T rue, and 0 corresponds to False.

Examples

```
=ISSTRING(VALUE("340")) -> 0
=ISSTRING(LEFT("Goodbye!",4)) -> 1
=ISSTRING("Some text") -> 1
```

@N

Summary Returns the given expression if it is a number , 0 otherwise.

Usage @N(*exp*)

Arguments *exp* an expression

Description If the given expression is a number , @N returns the expression, otherwise @N returns 0. @N is guaranteed to return a number and is used in functions that require numbers as parameters.

Examples

```
@N(@STRING(300,0)) -> 0
@N(34) -> 34
```

NEXT or @NEXT

Summary Returns the next logical value for a given value

Usage NEXT(*value* [, *inc*])

Arguments

value a number , string, or cell reference
inc the [optional] number by which to increment the value

Description The NEXT function returns the next logical value after *value* . The value is incremented by 1 unless an optional increment *inc* is specified. For example, if *value* is “Mon”, then NEXT(*value*) returns “T ue”. Values for months, days, and quarters are built in. If *value* is a number , NEXT adds 1 or *inc* to the number . For quarterly information, it will increment the quarter . When the quarter reaches 4, the next quarter is Q1 of the following year . For example, NEXT(“Q4 ’92”) is “Q1 ’93”.

Examples

```
=NEXT("Q4 '92") -> Q1 '93
=NEXT("January") -> February
=NEXT("Region 1") -> Region 2
```

NOT or @NOT

Summary Logical NOT .

Usage NOT(*num*)

Arguments

num any number

Description If *num* is zero or not a number , the NUM() function returns 1; otherwise, it returns 0.

Examples

```
=NOT(34) -> 0
=NOT("Text") -> 1
```

RANGE or @RANGE

Summary Gives the address for a numerically specified range.

Usage RANGE(*ur, lc, lr, rc*)

Arguments

ur	number corresponding the upper row of the range
lc	number corresponding the left column of the range
lr	number corresponding the lower row of the range
rc	number corresponding the right column of the range

Description RANGE returns the address for the range specified by the upper row , left column, lower row , right column. RANGE(0,0,1,1) is the range A1:B2. This can be useful in MScript scripts.

Examples

```
A1 : 0
B1 : 1
A2 : 5
B2 : 7
=SUM(RANGE(0,0,1,1)) is 13. This is the sum of the
values in the range A1:B2
=SUM(RANGE(A1,A1,B1,B1)) is also 13. Because cell
A1 contains 0 and cell B1 contains 1, this is the
same as the above example.
```

ROWS or @ROWS

Summary Returns the number of rows in a range.

Usage ROWS(*range*)

Arguments *range* the address of a worksheet range

Description This function counts the number of rows in the range *range* and returns the number .

Examples

```
=ROWS(A1:H20) -> 20
If RANGE1 is the name assigned to the range
B1:CC200, =ROWS(RANGE1) -> 200
```

@S

Summary Returns the given expression if it is a string, or a zero length string otherwise.

Usage @S(*exp*)

Arguments *exp* an expression

Description If the given expression is a string, this function returns the expression; otherwise it returns a zero length string. This function is used to guarantee that an expression is a string.

Examples

```
@S(343) -> "" (blank string)

A1: 2000
A2: Orders
@S(A1)&@S(A2) = Orders
```

SAME or @SAME

Summary Executes the same function as in the given cell

Usage SAME(*adr*)

Arguments

adr the address of the cell containing the desired formula

Description Takes the formula in cell *adr* and executes it as if it had been copied and then pasted into the current cell. This is a powerful and valuable function. You can have a master function and a series of functions that are the same. Where you change the master function, the method of calculation used by the other functions is changed. You can use SAME() in conjunction with INDEX(), HLOOKUP(), VLOOKUP(), and to define a formula used in a calculation rather than having a complex IF() statement. An application for this is an employee pay calculation that depends on an employee type.

Examples

```
A1: 1
A2: 2

If B1: =A1 * 5 and B2: =SAME(B1), then B2 = 10
(A2*5).
If B1: =A1/5, then B2 changes to 0.4 (A2/5)
If B1: =SIN(A1), B2 changes to 0.9093 (SIN(A2))
```

SETALTCOLOR or @SETALTCOLOR

Summary

Returns the given value and sets the alternate color to the specified color .

Usage

SET ALTCOLOR(*exp, colorconstant [, pattern]*), or

SET ALTCOLOR(*exp, r, g, b [, pattern]*)

Arguments

exp	any expression
colorconstant	any color constant
r	a number from 0 to 1, indicating the amount of red
g	a number from 0 to 1, indicating the amount of green
b	a number from 0 to 1, indicating the amount of blue

Description

Sets the alternate color of the current cell and returns the given expression. *Colorconstant* is one of the colors listed in the "Constants" section (see "Color Constants" on page 301), or *r, g*, and *b* are numbers between 0 and 1 that set the percentage of red, green, and blue that will be mixed together to create the color. White is all ones and black is all zeros. *Pattern* is a number greater than or equal to 0, referring to one of the patterns available in the Patterns Panel. If any of the color components or the pattern number are out of bounds or result in errors, the color is not changed.

Examples

```
=SETALTCOLOR(A1,IF(A1 < -100,1,0),0,0) : This sets
the alternate color to red if the number is less
than
-100, otherwise it sets the alternate color to
black.
=SETALTCOLOR(A1,MAGENTACOLOR) : This sets the
alternate color of cell A1 to Magenta.
```

SETBKGCOLOR or @SETBKGCOLOR

Summary

Returns the given value and sets the background color to the specified color .

Usage

SETBKGCOLOR(*exp, colorconstant [,pattern]*), or

SETBKGCOLOR(*exp, r, g, b [,pattern]*)

Arguments

exp any expression
colorconstant any color constant
r a number from 0 to 1, indicating the amount of red
g a number from 0 to 1, indicating the amount of green
b a number from 0 to 1, indicating the amount of blue

Description

Sets the background color of the current cell and returns the given expression. *Colorconstant* is one of the colors listed in the "Constants" section (see "Color Constants" on page 301), or *r, g*, and *b* are numbers between 0 and 1 that set the percentage of red, green, and blue that will be mixed together to create the color. White is all ones and black is all zeros. *Pattern* is a number greater than or equal to 0, referring to one of the patterns available in the Patterns Panel. If any of the color components or the pattern number are out of bounds or result in errors, the color is not changed.

Examples

```
=SETBKGCOLOR(A1,IF(A1 > 50 && A1 < 100,1,0),0,0) :
This sets the background color to red if the num-
ber is greater than 50 and less than 100, other-
wise it sets the background color to black.
=SETBKGCOLOR(A1,CYANCOLOR,56) : This sets the
background color of cell A1 to Cyan and the pat-
tern to pattern number 56.
```

SETCOLOR or @SETCOLOR

Summary

Returns the given value and sets the text color to the specified color .

Usage

SETCOLOR(*exp, colorconstant [,pattern]*), or

SETCOLOR(*exp, r, g, b [,pattern]*)

Arguments

exp any value
colorconstant any color constant
r a number from 0 to 1, indicating the amount of red
g a number from 0 to 1, indicating the amount of green
b a number from 0 to 1, indicating the amount of blue

Description

Sets the color of the current cell and returns expression. C*olorconstant* is one of the colors listed in the "Constants" section (see "Color Constants" on page 301), or *r, g,* and *b* are numbers between 0 and 1 that set the percentage of red, green, and blue that will be mixed together to create the color . White is all ones and black is all zeros. `Pattern` is a number greater than or equal to 0, referring to one of the patterns available in the Patterns Panel. If any of the color components or the pattern number are out of bounds or result in errors, the color is not changed.

Examples

```
=SETCOLOR(A1,IF(A1 > 100,1,0),IF(A1 > 100,0,1),0)
: This sets the color to red if the number is
greater than 100, otherwise it sets the color to
green.
=SETCOLOR(A1,BLUECOLOR) : This sets the color of
cell A1 to Blue.
```

SIGNAL or @SIGNAL

Summary

If the condition is true, sends range of cells to the named Mach port.

Usage

SIGNAL(*cond* ,*port* ,*range*)

Arguments

cond This function only sends data if this parameter is TRUE.

port A string containing the name of the port to send the data to.

range The range of cells to send to the named port.

Description

This function works in conjunction with the MesaListen class of objects to allow a spreadsheet to send a range of cells to a custom application in when a certain event takes place. The MesaListen object in the custom application can then take the appropriate action. An example is that a signal could be sent if an arbitrage opportunity presented itself. The parameters of the opportunity could be programmed into the worksheet. When the parameters became true, perhaps based of real time data being fed into the worksheet, the worksheet could send a signal to a custom application that would perform a securities trade.

Examples

```
=SIGNAL(GOLD_PRICE > 50 * SILVER_PRICE,"BUY_-
GOLD",A1) - send a message to the "BUY_GOLD"
object when the price of gold is 50 times higher
than the price of silver .
```

VLOOKUP or @VLOOKUP

Summary Looks up `code` in the vertical `table`.

Usage VLOOKUP(*code* , *table* , *offset*)

Arguments

code a string or number identifying the desired values
table the range of the lookup table
offset the offset of the relevant column

Description Returns the cell *offset* cells from the left side of the first match. If the first column contains only strings and code is a number , it returns the value from the last row . If *code* is also a string, but doesn't match any of the table entries, !ERR is returned.

Examples

```
     A    B
1    A    5.0
2    B    4.0
3    C    3.0
4    D    2.0
5    F      0

=VLOOKUP("A",A1:B5,1) -> 5.0
=VLOOKUP("D",A1:B5,1) -> 2.0
=VLOOKUP("Incomplete",A1:B5,1) -> !ERR
=VLOOKUP(3.0,A1:B5,1) -> 0
```

Date Functions

CONVDATE or @CONVDATE

Summary Converts a number in the format YYMMDD to a date serial number .

Usage CONVDA TE(*date*)

Arguments *date* a date given in YYMMDD format

Description Mesa stores dates in terms of date serial numbers that can be used in various date functions, and then translated back into the desired format. Dates after the year 2000 can be specified by placing a 1 before the year .

Examples

```
=CONVDATE(930210) -> 8441 : This is the date-num-
ber of 10-Feb-93
=CONVDATE(1100410) -> 14709 : This returns the
date number of 10-Apr-2010.
```

DATE or @DATE

Summary Returns a date serial number for the year , month and day .

Usage DATE(*yr, mon, day*)

Arguments

yr the year since 1900
mon the month
day the day of the month

Description A date serial number is a number used by Mesa to represent dates so that they may be formatted in a number of different ways. The number corresponds

to the number of days since January 1, 1970. Dates after the year 2000 can be specified by placing a 1 before the year .

Examples

```
=DATE(92,8,24) -> 8271 : This is the date number
for
24-Aug-92.
=DATE(79,10,31) -> 3590 : This is the date-number
for
31-Oct-79.
```

DATEVALUE, DAYS or @DATEVALUE

Summary

Datevalue converts a date string to a date serial number . If the value is a number it just returns the number .

Usage

DATEVALUE(*dtstr*), DATEVALUE(*num*)

Arguments

dtstr a string corresponding to a date
num any number

Description

This function converts dates entered as strings into the date serial number used by Mesa' s date functions. This function will not effect numeric input.

Examples

```
=DATEVALUE("02/04/92") = 8069
A3: 'Jul-94
=DATEVALUE(A3) = 8947 : This is the date number
for
1-Jul-94
```

DAY or @DAY

Summary Returns the day of the month for a date serial number.

Usage DAY(*date*)

Arguments *date* a date serial number

Description DAY returns the day of the month for a date serial number *date* . The date can also be a reference to a cell containing a date.

Examples

```
@day(9131) -> 4
A3: 18-Jul-92
=DAY(A3) -> 18

=DAY(NOW)  :  This formula will return the current
day of the month.
```

HOUR or @HOUR

Summary Returns the hour for a date serial number .

Usage HOUR(*date*)

Arguments *date* a date serial number

Description HOUR returns the hour for a date serial number *date* . The date can also be a reference to a cell containing a date. If a time is specified in a cell, but no date is specified, the integer portion of the date serial number is assumed to be zero.

Examples

```
=HOUR(TIMEVALUE("23:00:00")) -> 23
=HOUR(NOW) : This formula will return the current
hour of day.
```

MINUTE or @MINUTE

Summary Returns the minute for a date serial number .

Usage MINUTE(*date*)

Arguments *date* a date serial number

Description MINUTE returns the minutes from a date serial number *date* . The date can also be a reference to a cell containing a date. If a time is specified in a cell, but no date is specified, the integer portion of the date serial number is assumed to be zero.

Examples

```
=MINUTE(TIMEVALUE("11:35:20")) -> 35
=MINUTE(NOW) : This formula will return the cur-
rent number of minutes past the hour.
```

MONTH or @MONTH

Summary Returns the month for a date serial number .

Usage MONTH(*date*)

Arguments *date* a date serial number

Description MONTH returns the month of the year for a date serial number *date* . The date can also be a reference to a cell containing a date.

Examples

```
=MONTH(DATEVALUE("12/24/92")) -> 12

=CHOOSE(MONTH(NOW)-1,"Jan","Feb","Mar","A-
pr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec
")  :  This formula will display the current month.
```

NETWORKDAYS or @NETWORKDAYS

Summary Returns the number of working days between the given dates.

Usage NETWORKDA YS(*date1* , *date2* [,*holidays*])

Arguments

date1	the starting date serial number
date2	the ending date serial number
holidays	the serial numbers of any intermediate holidays

Description This function allows you to determine the number of working days between the two dates *date1* and *date2* . You can optionally list any holidays that fall between the two dates, which will not be counted as working days.

Examples

```
=NETWORKDAYS(NOW,NOW+10) : This returns the number
of working days between now and 10 days from now.
```

NEXTMONTH or @NEXTMONTH

Summary Returns a date serial number one calendar month after date.

Usage NEXTMONTH(*date*)

Arguments *date* a date serial number

Description This function returns the date serial number of the date one month from the given date. If the day of the month is greater than 28, then it is set to 28.

Examples

```
A3: 27-Jul-92
=NEXTMONTH(A3) -> 8274 : This is the date-number
for
27-Aug-92.

=NEXTMONTH(11381) -> 11409 : This is the date-num-
ber for 28-Mar-2001.
```

NEXTYEAR or @NEXTYEAR

Summary Returns the next year of the date.

Usage NEXTYEAR(*date*)

Arguments *date* a date serial number

Description This function returns the date serial number of the date one year from the given date. If the date is 2/29 of a leap year , the date returned is 2/28 of the following year .

Examples

```
=NEXTYEAR(NOW) : This formula returns a date-number one year from the current date.
=NEXTYEAR(DATE(93,6,20)) -> 8936 : This is the date-number for 20-Jun-94.
```

SECOND or @SECOND

Summary Returns the seconds of a date serial number .

Usage SECOND(*date*)

Arguments *date* a date serial number

Description SECOND returns the seconds from a date serial number *date* . The date can also be a reference to a cell containing a date. If a time is specified in a cell, but no date is specified, the integer portion of the date serial number is assumed to be zero.

Examples

```
=SECOND(TIMEVALUE("10:31:19")) -> 19
=SECOND(0.26903) -> 24 : This formula returns the seconds for the date-value for 06:27:24 AM.
```

TIME or @TIME

Summary Returns a date serial number for a given time.

Usage TIME(*hr, min, sec*)

Arguments

hr	the hour , from 0 to 23
min	the minutes
sec	the seconds

Description This function takes the time, for a given 24 hour time into the corresponding date serial number used by Mesa' s date functions.

Examples

```
=TIME(11,30,00) -> 0.479167 : This is the date-
number for 11:30 AM.
=NOW+TIME(8,0,0) : This formula returns a date-
number for the time 8 hours from now.
```

TIMEVALUE or @TIMEVALUE

Summary Converts a string to a date serial number .

Usage TIMEV ALUE(*time-str*)

Arguments *time-str* a string corresponding to a time

Description This function converts time entered as string into the date serial number used by Mesa' s date functions.

Examples

```
A1: '10
A2: '58
A3: '29
=TIMEVALUE(A1&":"&A2&":"&A3) = 0.45728 : This for-
mula takes the string contents of cells A1, A2 & A3
and concatenates them into the string "10:58:29",
then converts that string to a time value.

A3: 12:34 am
=TIMEVALUE(A3) -> 0.023611 : This is the date-num-
ber for 12:34 am.
```

WEEKDAY or @WEEKDAY

Summary Returns the day of the week for a date serial number , 0 - Sunday through 6 - Saturday .

Usage WEEKDA Y(*date*)

Arguments *date* a date serial number

Description This function is useful in determining what day of the week a given date serial number is.

Examples

```
=WEEKDAY(NOW-5) : This returns the day of the week
5 days prior to today.

A7: "6/20/82"
=WEEKDAY(DATEVALUE(A7)) -> 0 : Returns the day of
the week for 20-Jun-82.
```

WORKDAY or @WORKDAY

Summary Returns the date serial number a given number of working days from the `date` serial number . Working days are all days other than Saturday , Sunday , and the optional `holidays` listed.

Usage WORKDA Y(*date, days [,holidays]*)

Arguments

date a date serial number
days any number of days
holidays the serial number of any intermediate holidays

Description This function takes a date serial number and determines the date serial number of the day a given number of working days from then. If there are any holidays listed, they will not be counted as working

days by this function. Y ou may list any combination of dates, cell addresses containing dates, and ranges containing dates as arguments for *holidays* .

Examples

```
=WORKDAY(NOW,10) : This returns the date serial
number 10 working days from now.
```

YEAR or @YEAR

Summary Returns the year for a date serial number .

Usage YEAR(*date*)

Arguments *date* a date serial number

Description YEAR returns the year for a date serial number . The date can also be a reference to a cell containing a date.

Examples

```
=YEAR(NOW-1000) : This returns the year of the
date 1,000 days prior to the current date.

A7: '6/20/82
=YEAR(DATEVALUE(A7)) -> 82 : This formula extract
the year from the date-string "6/20/82".
```

Database Functions

For more information about criteria ranges, see “Criteria Ranges” on page 128.

DAVE, or DAVERAGE

Summary

Takes the average of numeric records that match criteria.

Usage

DAVE(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the column to be averaged
crit	the criteria range

Description

This function takes a range and averages all the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column.

Examples

```
This is a sample database:
       A      B     C
1    NAME   GRADE  GPA    GRADE
2    Joe    B       3.2   B
3    Sue    A       4.7
4    Robert C       2.7
5    Steve  A       4.5
6    Ann    B       4.2

The database range is A1:C6. The criteria range is
D1:D2.

To calculate the average GPA of everyone who
earned a B use this formula:

=DAVE(A1:C6,2,D1:D2) -> 3.7
```

@DAVE or @DAVG

Summary

Takes the average of numeric records that match criteria. Strings have a value of zero.

Usage

@DAVE(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the column to be averaged
crit	the criteria range

Description

This function takes a range and averages all the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column. This function is basically the same as DA VE() or DA VERAGE() except that those function ignore string values.

Examples

```
This is a sample database:
       A       B      C
1    NAME    GRADE  GPA     GRADE
2    Joe     B       3.2    B
3    Sue     A       4.7
4    Robert  C       2.7
5    Steve   A       4.5
6    Ann     B       4.2

The database range is A1:C6. The criteria range is
D1:D2.

To calculate the average GPA of everyone who
earned a B use this formula:

@DAVE(A1:C6,2,D1:D2) -> 3.7
```

DCOUNT or @DCOUNT

Summary

Counts all the numeric items matching criteria in the column offset from the upper left corner of the database range.

Usage

DCOUNT(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and counts all the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column.

Examples

```
This is a sample database:
       A      B     C       D
1    NAME   GRADE  GPA    GRADE
2    Joe    B       3.2   B
3    Sue    A       4.7
4    Robert C       2.7
5    Steve  A       4.5
6    Ann    B       4.2

The database range is A1:C6. The criteria range is
D1:D2.

To count the number of students who earned B's use
this formula:

=DCOUNT(A1:C6,1,D1:D2) -> 2
```

DMAX

Summary

Returns the maximum numeric value of the records in database that match criteria.

Usage

DMAX(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and finds the maximum number of the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column.

Examples

```
This is a sample database:
       A       B      C         D
1    NAME    SALES  REGION   REGION
2    Joe      3000  East     East
3    Sue      4500  North
4    Robert   2800  East
5    Steve    3700  South
6    Ann      2400  West

The database range is A1:C6.  The criteria range
is D1:D2.

To find the largest sales figure in the Eastern
region use this formula:
=DMAX(A1:C6,1,D1:D2) = 3000
```

@DMAX

Summary

Returns the maximum value of the records in database that match criteria. Strings have a value of zero.

Usage

@DMAX(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and finds the highest number of the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column. Strings are assumed to have a value of zero. This function is basically the same as DMAX except that those function ignore string values

Examples

```
This is a sample database:
       A       B      C         D
1    NAME    SALES  REGION   REGION
2    Joe      3000  East     East
3    Sue      4500  North
4    Robert   2800  East
5    Steve    3700  South
6    Ann      2400  West

The database range is A1:C6. The criteria range is
D1:D2.

To find the largest sales figure in the Eastern
region use this formula:
@DMAX(A1:C6,1,D1:D2) -> 3000
```

Note: DMAX() and @DMAX differ in that DMAX ignores string values and @DMAX treats strings as having a value of zero.

DMIN

Summary

Returns the smallest numeric item in the records that match the criteria.

Usage

DMIN(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and finds the minimum number of the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column.

Examples

```
This is a sample database:
       A       B      C         D
1    NAME    SALES  REGION   REGION
2    Joe      3000  East     East
3    Sue      4500  North
4    Robert   2800  East
5    Steve    3700  South
6    Ann      2400  West

The database range is A1:C6. The criteria range is
D1:D2.

To find the smallest sales figure in the Eastern
region use this formula:
=DMIN(A1:C6,1,D1:D2) = 2800
```

@DMIN

Summary

Returns the smallest numeric item in the records that match the criteria. String values will count as zeros.

Usage

@DMIN(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and finds the minimum number of the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column. Strings are assumed to have a value of zero. This function is basically the same as DMIN except that those function ignore string values

Examples

```
This is a sample database:
       A       B      C         D
1    NAME    SALES  REGION   REGION
2    Joe      3000  East     East
3    Sue      4500  North
4    Robert   2800  East
5    Steve    3700  South
6    Ann      2400  West

The database range is A1:C6.  The criteria range
is D1:D2.

To find the smallest sales figure in the Eastern
region use this formula:
@DMIN(A1:C6,1,D1:D2) -> 2800
```

Note: DMIN() and @DMIN() differ in that DMIN() ignores strings, while @DMIN() treats strings as having a value of zero.

DPROD

Summary

Returns the product of all the items matching the criteria in the column offset from the upper left corner of the range. Strings are ignored.

Usage

DPROD(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and multiplies all the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column.

Examples

```
This is a sample database:
      A      B         C         D
1   GATE   PATH   % PASSING    PATH
2   1      North       0.96    East
3   2      South       0.95
4   3      West        0.97
5   4      East        0.95
6   5      East        0.93
7   6      North       0.88
8   7      West        0.92
9   8      East        0.99
10  9      South       0.94

The database range is A1:C10. The criteria range
is D1:D2.

To find the product of the percentages for the
East path use this formula:
=DPROD(A1:C10, 2, D1:D2) -> 0.874665
```

@DPROD

Summary

Returns the product of all the items matching the criteria in the column offset from the upper left corner of the range. Strings are treated as zero values.

Usage

@DPROD(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and multiplies all the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column. Strings are treated as

having a value of zero. This function is basically the same as DPROD except that those function ignore string values

Examples

```
This is a sample database:
      A       B          C          D
1   GATE    PATH    % PASSING     PATH
2   1       North         0.96    East
3   2       South         0.95
4   3       West          0.97
5   4       East          0.95
6   5       East          0.93
7   6       North         0.88
8   7       West          0.92
9   8       East          0.99
10  9       South         0.94

The database range is A1:C10.  The criteria range
is D1:D2.

To find the product of the percentages for the
East path use this formula:
@DPROD(A1:C10, 2, D1:D2) -> 0.874665
```

Note: DPROD() and @DPROD differ in that DPROD() ignores string values, but @DPROD() treats strings as having a value of zero.

DSTDDEV or DSTDEV

Summary

Returns the standard deviation of the numeric records that match the criteria.

Usage

DSTDEV(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and finds the standard deviation of the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column.

Examples

```
This is a sample database:
      A        B        C        D
1   STATE   REGION   AVG RF   REGION
2   AL      South       3.7   West
3   CA      West        2.1
4   NJ      East        5.2
5   FL      South       4.9
6   LA      South       6.8
7   TX      West        5.0
8   MA      North       7.6
9   AZ      West        2.2
10  NV      West        2.4

The database range is A1:C10. The criteria range
is D1:D2.

To find the standard deviation of rainfall in the
Western region use this formula:
=DSTDEV(A1:C10,2,D1:D2) -> 1.2029
```

@DSTD

Summary

Returns the standard deviation of the records that match the criteria. Strings have a value of zero.

Usage

@DSTD(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and finds the standard deviation of the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column. Strings are assumed to have a value of zero. This function is basically the same as DSTDEV except that those function ignore string values

Examples

```
This is a sample database:
       A       B      C         D
1    NAME    SALES  REGION   REGION
2    Joe      3000  East     East
3    Sue      4500  North
4    Robert   2800  East
5    Steve    3700  South
6    Ann      2400  West
The database range is A1:C10. The criteria range
is D1:D2.

To find the standard deviation of rainfall in the
Western region use this formula:
@DSTD(A1:C10,2,D1:D2) -> 1.2029
```

DSUM or @DSUM

Summary

Sums all the items matching criteria in the column offset from the upper left corner of the database range.

Usage

DSUM(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and sums all the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column.

Examples

```
This is a sample database:
      A        B        C        D
1   STATE   REGION   AVG RF   REGION
2   AL      South       3.7   South
3   CA      West        2.1
4   NJ      East        5.2
5   FL      South       4.9
6   LA      South       6.8
7   TX      West        5.0
8   MA      North       7.6
9   AZ      West        2.2
10  NV      West        2.4

The database range is A1:C10. The criteria range
is D1:D2.

To find the total amount of rainfall for the
Southern region use this formula:
=DSUM(A1:C10,2,D1:D2) -> 15.4
```

DVAR

Summary

Returns the variance of the numeric records that match the criteria. Strings are ignored.

Usage

DVAR(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and finds the variance of the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column.

Examples

```
This is a sample database:
      A        B        C        D
1   STATE   REGION   AVG RF   REGION
2   AL      South       3.7   West
3   CA      West        2.1   South
4   NJ      East        5.2
5   FL      South       4.9
6   LA      South       6.8
7   TX      West        5.0
8   MA      North       7.6
9   AZ      West        2.2
10  NV      West        2.4

The database range is A1:C10. The criteria range
is D1:D2.

To find the variance of rainfall in the Southern
and Western regions use this formula:
=DVAR(A1:C10,2,D1:D3) -> 2.7192
```

@DVAR

Summary

Returns the variance of the records that match the criteria. Strings have a value of zero.

Usage

@DVAR(*database, offset, crit*)

Arguments

database	the range address of the database
offset	the offset of the relevant column
crit	the criteria range

Description

This function takes a range and finds the variance of the numbers in the column at the offset value from the first column of the range that meet the criteria defined in another column. Strings are treated as having a value of zero. This function is basically the same as DV AR except that those function ignore string values

Examples

```
This is a sample database:
      A        B        C        D
1   STATE   REGION   AVG RF   REGION
2   AL      South       3.7   West
3   CA      West        2.1   South
4   NJ      East        5.2
5   FL      South       4.9
6   LA      South       6.8
7   TX      West        5.0
8   MA      North       7.6
9   AZ      West        2.2
10  NV      West        2.4
The database range is A1:C10. The criteria range
is D1:D3.

To find the variance of rainfall in the Southern
and Western regions use this formula:
@DVAR(A1:C10,2,D1:D3) -> 2.7192
```

Note: DVAR() and @DV AR() are different in that DVAR() ignores string values, while @DV AR treats strings as having a value of zero.

Constants

Color Constants

Color constants are used by NEXTSTEP to specify how the amount of red, blue, and green present in a given color. Color constants can be used in Mesa's functions to manipulate the color of cells and ranges. Here is a list of the available color constants.

- BLACKCOLOR, @BLACKCOLOR
- BLUECOLOR, @BLUECOLOR
- CYANCOLOR, @CY ANCOLOR
- DKGRA YCOLOR, @DKGRA YCOLOR
- GREENCOLOR, @GREENCOLOR
- LTGRA YCOLOR, @L TGRA YCOLOR
- MAGENT ACOLOR, @MAGENT ACOLOR
- REDCOLOR, @REDCOLOR
- WHITECOLOR, @WHITECOLOR
- YELLOWCOLOR, @YELLOWCOLOR

Format Constants

Mesa uses the following format constants in its formatting functions. These formats correspond to the different cell style formats in the Cell Style Inspector. (See "Cell Style Inspector" on page 75.)

- GENERALFORMA T, @GENERALFORMA T: This constant is used to specify the general format.

```
=FORMAT(238.093, GENERALFORMAT) -> 238.093
```

- FIXEDFORMA T, @FIXEDFORMA T: This constant is used to specify the fixed decimal format.

```
=FORMAT(42.692, FIXEDFORMAT) -> 42.69
```

- SCIENTIFICFORMA T, @SCIENTIFICFORMA T: This constant is used to specify the scientific notation format.

```
=FORMAT(1.69, SCIENTIFICFORMAT) -> 1.89e+01
```

- CURRENCYFORMA T, @CURRENCYFORMA T: This constant is used to specify the currency format.

```
=FORMAT(555.9, CURRENCYFORMAT) -> $555.90
```

- COMMAFORMA T, @COMMAFORMA T: This constant is used to specify the comma format.

```
=FORMAT(238.093, COMMAFORMAT) -> 238.09
```

- CHARTFORMA T, @CHARTFORMA T: This constant is used to specify the chart format.

```
=FORMAT(7, CHARTFORMAT) -> +++++++
```

- PERCENTFORMA T, @PERCENTFORMA T: This constant is used to specify the percent format.

```
=FORMAT(0.09389, PERCENTFORMAT, 2) -> 9.39%
```

- DMYFORMA T, @DMYFORMA T: This constant is used to specify the date day-month-year format.

```
=FORMAT(9876.54321, DMYFORMAT) -> 15-Jan-97
```

- DAYMONFORMA T, @DAYMONFORMA T: This constant is used to specify date day-month format.

```
=FORMAT(9876.54321, DAYMONTHFORMAT)
-> 15-Jan
```

- MONYEARFORMA T, @MONYEARFORMA T: This constant is used to specify date month-year format.

```
=FORMAT(9876.54321, MONYEARFORMAT)
-> Jan-97
```

- HMSFORMA T, @HMSFORMA T: This constant is used to specify date hours-minutes-seconds format.

```
=FORMAT(9876.54321, HMSFORMAT)
-> 1:02:13 PM
```

- HOURMINFORMA T, @HOURMINFORMA T: This constant is used to specify date hours-minutes format.

```
=FORMAT(9876.54321, HOURMINFORMAT)
-> 1:02 PM
```

- DINTL1FORMA T, @DINTL1FORMA T: This constant is used to specify the numeric date month-day-year format.

```
=FORMAT(9876.54321, DINTL1FORMAT)
-> 01/15/97
```

- DINTL2FORMA T, @DINTL2FORMA T: This constant is used to specify the numeric date month-day format.

```
=FORMAT(9876.54321, DINT2FORMAT) -> 01/15
```

- TINTL1FORMA T, @TINTL1FORMA T: This constant is used to specify twenty four hour time format with seconds.

```
=FORMAT(9876.54321, TINTL1FORMAT)
 -> 13:02:13
```

- TINTL2FORMA T, @TINTL2FORMA T: This constant is used to specify twenty four hour time format without seconds.

```
=FORMAT(9876.54321, TINTL2FORMAT) -> 13:02
```

Math Constants

Mesa uses the following mathematical constants in its functions.

- PI, @PI: The area of a circle of radius one, approximately 3.14159.
- RANDOM, RAND, @RAND: This function returns a random number between zero and one, including zero and excluding one.

Date Constants

Mesa uses the following date constants in its date functions to specify current date or time.

- NOW, @NOW: Returns the current date serial number .
- TODAY, @TODAY: Returns the current date serial number without the current time, so that the serial

number only contains the day month and year information.

Special Constants

- CURRENTNUMBER or @CURRENTNUMBER: The current numeric value of a cell. Can be used for time series functions. To reset a cell containing this constant, use **Edit → Clear Special → Reset.**

```
To remember the largest number entered into
   cell A1, use
=if(a1 > currentnumber,a1,currentnumber).

To count the number of times the worksheet
   was recalculated, use
currentnumber + 1
```

- CURRENTVALUE or @CURRENTVALUE: The current value of the cell (before the recalc). This is useful for comparing old cell values with other values, such as taking the larger of the current value or the value of another cell.To reset a cell containing this constant, use **Edit → Clear Special → Reset.**
- ERR or @ERR: Returns an error. All cells that refer to a cell with an error will evaluate to errors as well.
- ERRORCONST: Returns an error. All cells that refer to a cell with an error will evaluate to errors as well.
- FALSE(), FALSE or @FALSE: Returns a logical false value.
- FILENAME or @FILENAME: Returns the filename of the current worksheet.
- HOSTNAME, or @HOSTNAME: Returns the name of the machine that Mesa is running on.
- NA or @NA: Returns a special error, NA. All cells that refer to a cell with an error will evaluate to errors as well. @NA is useful to designate cells that have a value that is not available or defined.

- NACONST : Returns a special error , NA. All cells that refer to a cell with an error will evaluate to errors as well. NACONST is useful to designate cells that have a value that is not available or defined.
- PATHNAME or @P ATHNAME: Returns the path of the current worksheet.
- THISCOLUMN: Returns the column number of the cell containing this constant.
- THISROW : Returns the row number of the cell containing this constant.
- TRUE(), TRUE or @TRUE: Returns a logical true value.
- USERNAME, or @USERNAME: Returns the user 's login name.

8. MScript

The Language

MScript is a scripting language designed for the Mesa user. It is a very simple and straight-forward language that is similar to BASIC. In short, it was designed for ease of use by people who aren't programmers. MScript is ideal for writing simple scripts and macros, but AddIns or MOLI should be used for more complex tasks. See the *Mesa Programmer's Guide* for more information about AddIns and MOLI Note that Addins and MOLI are not available in Sunrise.

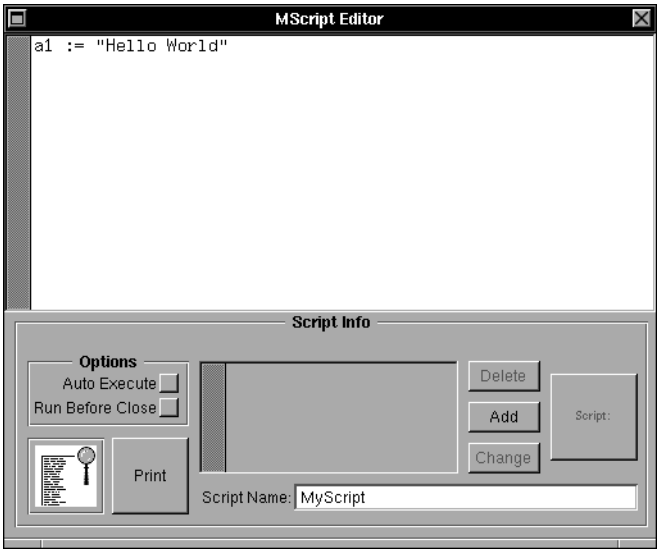
Creating and Editing a Script

MScript scripts can be created and edited using the MScript Inspector. The MScript Inspector can be opened with the **Sheet** → **MScript Inspector...** command.

Creating a Script

You can type MScript commands directly into the editing window of the MScript Inspector. For example, type:

```
a1 := "Hello World"
```



A script needs a name before it can be added and run. Type `MyScript` into the field marked "Script Name:" and click the *Add* button. Choose *MyScript* from the browser. Click on the *Script:MyScript* button to execute your script.

Note: The *Script:MyScript* button can be *Command-dragged* to one of the Tool Holders on your worksheet or onto the worksheet itself, if you so desire.

Editing a Script

To change a script, select the appropriate script name from the browser and make your changes in the editing window. When you've finished your editing, click on the *Change* button and run your script as normal.

Errors

If there's an error in your script, a message will appear below the editing window, and the line containing the error will be highlighted.

Basic Operations

Each statement in an MScript script must be on its own line.

Comments can be placed in scripts by starting the line with a `%`. Comments can be useful for making your scripts more readable and more easily maintainable by others and by the author at a later date.

```
% This line is a comment and won't affect
% the script.

a1 := "Hello World"

% The above line of code will put the string
% "Hello
%  World" in the cell A1.
```

Manipulating Values

The values in cells can be set easily from an MScript script. Numbers, strings, cell references, range labels, and formulas must be treated differently.

```
% This will set the value of A1 to the
% number 10.
a1 := 10

% This will set the value of A1 to the
% string "Hello"
% Note that strings need to be surrounded by
% quotes.
a1 := "Hello"

% This will set the value of A1 to the value
%  of B1
a1 := b1

% This will set the value of A1 to the value
%  of the range label "mylabel". If there
%  is no range labelled "mylabel", an error
%  will occur.
a1 := mylabel

% This will set the input string of A1 to
%  the formula "=1+2"
% Note that strings that should be
%  interpreted as if they were typed in by
%  a user must be treated in this manner.
setinput(a1,"=1+2")
```

For more information about **setinput()**, see “SETINPUT” on page 377.

Variables

Variables can be used to facilitate the manipulation of information within an MScript script. Variable names must be preceded by a `?`, and can contain upper- and lowercase letters (MScript variables are case-insensitive), numbers, or the symbols “`_`”, “`$`”, “`?`” and “`.`”. Examples of valid MScript variable names include `?x`, `?value37`, `?cat`, `?tax_rate`, and `?overtime.hours`.

Variables can have string or number values, cell references, or labels assigned to them, and can be used in the place of those values or references. Values can be assigned to variables with the `LET()` function or with the `:=` operator. For example:

```
% This sets the variable ?x to the value 10
LET("?x",10)

% This does the same thing.
?x := 10

% This sets the variable ?z to the string
%  "hello"
?z := "hello"
```

Variables can also be set to the values contained in a cell. When a variable is set to the value of a cell, the variable's value will change as the value of the cell changes. In order to set a variable to the current value of the cell (such that its value will not change), use the **freeze()** function. For example:

```
% This sets the variable ?y to the value in
%  cell A1
% If the value in cell A1 changes, the value
%  of ?y will
%  change as well.
?y := a1

% This sets the variable ?y to the current
%  value in the
%  cell A1
% The value of ?y will not change if the
%  value in cell A1
%  changes
?y := freeze(a1)
```

Conditionals

IF-ELSE statements

One way to execute code conditionally is with an IF/ELSE statement. If the expression following the IF statement is true, the next line will executed. Otherwise the next statement is executed. The ELSE lines

are skipped if the IF expression was true. In the example below, since the condition is true, the next line is executed.

```
IF (1 = 1)
a1 := "One does in fact equal One"
ELSE
a1 := "Whoops..."
```

If you want to execute more than one function in an IF/ELSE statement, lines can be grouped after the IF and ELSE statements with BEGIN and END statements.

```
IF (1 = 1)
    BEGIN
    a1 := "One does in fact equal One"
    a2 := "And MScript tells us so!"
    END
ELSE
    BEGIN
    a1 := "Whoops..."
    a2 := "Something's wrong!"
    END
```

Conditionals can be used to check values and execute specific functions. More complex conditions can be checked using operators like the ones listed in the following table.

Conditional Operators

Symbol	Meaning
!=	not equal to
<	less than
<=	less than or equal
<>	not equal to
=	equal
=<	equal to or less than
==	equal
=>	equal to or greater than
>	greater than
>=	greater than or equal to
AND	logical AND
OR	logical OR
XOR	exclusive OR
^^	exclusive OR
||	logical OR

The following conditional statements are some simple examples:

```
if (a1 > 0 AND a1 < 10)
b1 := "A1 is between 0 and 10"

if (a2 < 10 OR a2 > 20)
b2 := "b1 is either less than 10 or greater
   than 20"

if (a3 = 10 XOR a4 > 10)
b3 := "Either a3 is equal to 10 or a4 is
   greater than 10, but not both."
```

WHILE statements

Another way to execute code conditionally is with a WHILE statement. A WHILE statement will execute the line following it as long as the expression following the WHILE keyword is true. In the following example, the value in cell b1 will be set to 10.

```
% make ?x equal to 1
?x := 1

% As long as ?x is less than 10, keep adding
%  1 to it.
WHILE ?x < 10
?x := ?x + 1

% set the value of the cell b1 to the value
%  of ?x
b1 := ?x
```

Lines after a WHILE statement can also be grouped with BEGIN and END.

Looping and Branching

Looping and branching can be accomplished with a GOTO statement. “GOTO *label* ” passes control unconditionally to another section of the script, marked by a *label* . Labels begin with a colon (:).

```
a1 := "This is an example of branching."

GOTO "mylabel"

a2 := "Program control never gets to this
   line."

:mylabel
a3 := "...but it does get here."
```

Looping can be accomplished with the use of a conditional and a GOTO. The example below will put the string “This is an example of looping” in cells A1:A20.

```
% set ?x to 0
?x := 0

% set a label
:mylabel
% Set the string value of the cell at row
%  ?x, column 0
address(?x,0) := "This is an example of
   looping."
% add 1 to ?x
?x := ?x + 1

% As long as x is less than 20, do it all
%  again.
if ?x < 20
   GOTO "mylabel"
```

For more information about the **address()** function, see “ADDRESS or @ADDRESS” on page 252.

Subroutines

Subroutines can be used to divide a large script up into smaller parts for the sake of readability and functionality. They can also be used to perform small tasks that have to be repeated numerous times.

Subroutines are called with "GOSUB *label* ". GOSUB is similar to GOTO, but will return control to the original location in the script upon reaching a RETURN statement.

```
% Check if A1 is 1, 2, or 3
if (a1 = 1 or a1 = 2 or a1 = 3)
begin
    % Concatenate a string for the appropri-
    ate
    % subroutine name
    gosub "subroutine" & fixed(a1,0)
    % Set cell B1 to the message set by the
    subroutine
    b1 := ?message
end
else
begin
    % Set cell B1 to an error message tell-
    ing the user
    % that they didn't pick one of our sub-
    routines
    b1 := "Not a valid subroutine."
end

:subroutine1
?message := "This is subroutine #1"
return

:subroutine2
?message := "This is subroutine #2"
return

:subroutine3
?message := "This is subroutine #3"
```

For more information about the **fixed()** function, see "FIXED or @STRING" on page 243.

Getting user input

MScript provides three different ways to get input from the user. The user can be prompted to type something, can be offered a menu with choices, or can be given an alert panel with buttons.

Prompting for input

The user can be prompted for input using the **prompt()** function. The **prompt()** function will cause a window to appear on the screen displaying the specified text, and will return the string typed in by the user.

```
% This will set the value of A1 to the
   string typed in by
%  the user.
a1 := prompt("Type in any string")
```

For more information on the **prompt()** function, see "PROMPT" on page 374).

Menus

The user can be prompted to make a choice from a menu using the **menu()** command. For a detailed description, see "MENU" on page 372.

If the worksheet contained the following information:

```
    A      B                    C
1   One    The First Option     subroutine1
2   Two    The Second Option    subroutine2
3   Three  The Third Option     subroutine3
```

then the following code would create a menu of the options and execute the appropriate subroutines.

```
?foo := menu(a1:c3,menubyrow)
IF (?foo != -1)
    BEGIN
    gosub ?foo
    e1 := ?message
    END
ELSE
    e1 := "No choice was made"

:subroutine1
?message := "This is subroutine #1"
return

:subroutine2
?message := "This is subroutine #2"
return

:subroutine3
?message := "This is subroutine #3"
```

Alert Panels

The user can also be prompted for input via an Alert panel with buttons. For more information about the **alert()** function, see “ALERT” on page 370.

```
?foo := alert("Options","Choose one.",
    "One","Two", "Three")
IF (?foo != -1)
BEGIN
    gosub "subroutine" & fixed(?foo+1,0)
    e1 := ?message
END
ELSE
    e1 := "No choice was made"

:subroutine1
?message := "This is subroutine #1"
return

:subroutine2
?message := "This is subroutine #2"
return

:subroutine3
?message := "This is subroutine #3"
```

Functions

MScript can use all Mesa functions, although some functions take different arguments in the scripting language. The following sections describe these differences as well as additional MScript-specific functions. For more information about standard Mesa functions, see “Functions” on page 186.

File Operations

NAMESHEET

Summary Names the current worksheet.

Usage NAMESHEET(*name*)

Arguments *name* the new name of the worksheet

Description This function changes the name of the current worksheet to *name* . The name *name* must be a string.

Returns the name of the sheet if success, -1 if failure

Examples

```
namesheet("MySheet")
```

SAVE

Summary Saves the current worksheet.

Usage SAVE([*name*])

Arguments *name* the [optional] new name of the worksheet

Description This function saves the current worksheet. If *name* is specified, the worksheet' s name is changed before the save. The name *name* must be a string.

Returns the full pathname of the sheet if success, -1 if failure

Examples

```
save()
```

```
save("NewName")
```

Editing Operations

CLR or CLEAR

Summary Clears the specified cell or range.

Usage CLR(*cell*) or CLR(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function clears all the information from the *cell* or *range* specified, including both data and formatting information.

Returns *cell* or *range*

Examples

```
clear(a1)
clear(a1:b7)
clear(myrange)
```

CLRDATA or CLEARDATA

Summary Clears any data from the specified cell or range.

Usage CLRDA TA(*cell*) or CLRDA TA(*range*)

Arguments
cell any cell reference
range any worksheet range

Description This function clears the data (but not the formatting information) from *cell* or *range* .

Returns *cell* or *range*

Examples

```
clrdata(a1)
clrdata(a1:b7)
clrdata(myrange)
```

CLRFMT, CLEARFMT or CLEARFORMAT

Summary Clears any formatting information from the specified cell or range.

Usage CLRFMT(*cell*) or CLRFMT(*range*)

Arguments
cell any cell reference
range any worksheet range

Description This function clears formatting information (but not the data) from *cell* or *range* .

Returns *cell* or *range*

Examples

```
clrfmt(a1)
clrfmt(a1:b7)
clrfmt(myrange)
```

CLRNUM or CLEARNUM

Summary Clears numerical data from the specified cell or range.

Usage CLRNUM(*cell*) or CLRNUM(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function clears the numerical input from *cell* or *range* , but not strings, formulas, or formatting information.

Returns *cell* or *range*

Examples

```
clrnum(a1)

clrnum(a1:b7)

clrnum(myrange)
```

CLRSTR or CLEARSTR

Summary Clears string data from the specified cell or range.

Usage CLRSTR(*cell*) or CLRSTR(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function clears any string data from *cell* or *range* , but not numerical input, formulas, or formatting information. It returns *cell* or *range* .

Returns *cell* or *range*

Examples

```
clrstr(a1)

clrstr(a1:b7)

clrstr(myrange)
```

COPY

Summary Copies a cell or range from one location to another .

Usage COPY(*source* ,*dest*)

Arguments

source	any cell reference or worksheet range
dest	any cell reference or worksheet range

Description The COPY function copies the information from *source* into *dest* . If a cell reference is given for *dest* , it will be used as the upper left hand corner of the destination range. Any information in the destination range will be overwritten.

Returns *dest*

Examples

```
copy(a1:b7,d12)

copy(myrange,d12:e14)
```

COPYFMT or COPYFORMAT

Summary

Copies formatting information of a cell or range from one location to another .

Usage

COPYFMT(*source* ,*dest*)

Arguments

source any cell reference or worksheet range
dest any cell reference or worksheet range

Description

This function copies formatting information only (but not the values) from *source* into *dest* . If a cell reference is given for *dest* , it will be used as the upper left hand corner of the destination range. Any information in the destination range will be overwritten.

Returns

dest

Examples

```
copyfmt(a1:b7,d12)

copyfmt(myrange,d12:e14)
```

COPYVAL

Summary Copies the values of a cell or range from one location to another .

Usage COPYVAL(*source* ,*dest*)

Arguments

source any cell reference or worksheet range
dest any cell reference or worksheet range

Description This function copies the values only (but not the formatting) from *source* into *dest* . If a cell reference is given for *dest* , it will be used as the upper left hand corner of the destination range. Any information in the destination range will be overwritten.

Returns *dest*

Examples

```
copyval(a1:b7,d12)

copyval(myrange,d12:e14)
```

FORMTOVAL or FORMULASTOVAL

Summary Converts formulas to their current values.

Usage FORMTOV AL(*cell*) or FORMTOV AL(*range*)

Arguments

cell any cell reference
range any worksheet range

Description This function converts any formulas in the *cell* or *range* to their current values. Any cell references within the formulas will be converted to the values of the cells referenced, and the values of the formulas will be calculated.

Returns *cell* or *range*

Examples

```
formtoval(a1)

formtoval(a1:b7)

formtoval(myrange)
```

LOWER

Summary Converts strings in a cell or range to lower case.

Usage LOWER(*cell*) or LOWER(*range*)

Arguments

cell any cell reference
range any worksheet range

Description Converts the strings in the *cell* or *range* to lower case, regardless of their original case.

Returns *cell* or *range*

Examples

```
lower(a1)

lower(a1:b7)

lower(myrange)
```

MOVE

Summary Moves the information in a cell or range.

Usage MOVE(*source* ,*dest*)

Arguments

source any cell reference or worksheet range
dest any cell reference or worksheet range

Description Moves the information from *source* into *dest* . If a cell reference is given for *dest* , it will be used as the upper left hand corner of the destination range. Any information in the destination range will be overwritten.

Returns *dest*

Examples

```
move(a1:b7,d12)

move(myrange,d12:e14)
```

NUMTOSTR

Summary Converts numbers in the specified cell or range to strings.

Usage NUMTOSTR(*cell*) or NUMTOSTR(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function any converts numerical values in *cell* or *range* to strings.

Returns *cell* or *range*

Examples

```
lower(a1)

lower(a1:b7)

lower(myrange)
```

PROPER

Summary Converts the strings in the specified cell or range to Proper Case.

Usage PROPER(*cell*) or PROPER(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function converts the strings in the *cell* or *range* to Proper Case, regardless of their original case. The first letter of each word is capitalized, and all remaining letters are converted to lower case.

Returns *cell* or *range*

Examples

```
proper(a1)

proper(a1:b7)

proper(myrange)
```

SELECT

Summary Selects a cell or range in the worksheet.

Usage SELECT(*cell*) or SELECT(*range* [, *selectedcell*])

Arguments

cell	any cell reference
range	any worksheet range
selectedcell	an [optional] cell reference within

Description This function highlights *cell* or *range* . If *selectedcell* is specified and is within *range* , that cell is selected; otherwise, the upper left cell is selected.

Returns *cell* or *range*

Examples

```
select(b17)
select(a10:g19,b13)
```

SELECTALL

Summary Selects all cells in the sheet.

Usage SELECT ALL([*selectedcell*])

Arguments

selectedcell	an [optional] cell reference

Description This function highlights the entire sheet. If *selected-cell* is specified, that cell is selected; otherwise, the upper left cell (A1) is selected.

Returns the range reference of the selected area

Examples

```
selectall()
selectall(g29)
```

STRTONUM

Summary Converts strings in the specified cell or range to numbers.

Usage STRTONUM(*cell*) or STRTONUM(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function converts any strings in *cell* or *range* to their numerical equivalent.

Returns *cell* or *range*

Examples

```
strtonum(a1)

strtonum(a1:b7)

strtonum(myrange)
```

TRANSPOSE

Summary Transposes the data from a range, switching rows and columns.

Usage TRANSPOSE(*source* ,*dest*)

Arguments

source	any cell reference or worksheet range
dest	any cell reference or worksheet range

Description This function transposes the information from *source* into *dest* , such that rows and columns are transposed. If a cell reference is given for *dest* , it will be

used as the upper left hand corner of the destination range. Any information in the destination range will be overwritten.

Returns *dest*

Examples

```
transpose(a1:b7,d12)
transpose(myrange,d12:e14)
```

UPPER

Summary Converts strings in the specified cell or range to UPPER CASE.

Usage UPPER(*cell*) or UPPER(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function converts any strings in the *cell* or *range* to UPPER CASE, regardless of their original case.

Returns *cell* or *range*

Examples

```
upper(a1)
upper(a1:b7)
upper(myrange)
```

Formatting Operations

BOLD

Summary Toggles the data in the specified range between **bold** and regular weight.

Usage BOLD(*cell*) or BOLD(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function changes the format of the selected cells to **bold** if they are regular weight, and to regular weight if they are **bold** .

Returns *cell* or *range*

Examples

```
bold(a1)

bold(a1:b7)

bold(myrange)
```

CLRHBREAK or CLEARHBREAK

Summary Clears horizontal page breaks from the specified cell or range.

Usage CLRHBREAK(*cell*) or CLRHBREAK(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function clears any horizontal page breaks within the *cell* or *range* , including breaks following the last cell in the range, but not breaks preceding the first cell.

Returns *cell* or *range*

Examples

```
clrhbreak(a1)

clrhbreak(a1:b7)

clrhbreak(myrange)
```

CLRVBREAK or CLEARVBREAK

Summary Clears horizontal page breaks from the specified cell or range.

Usage CLR VBREAK(*cell*) or CLR VBREAK(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function clears any vertical page breaks within the *cell* or *range* , including breaks following the last cell in the range, but not breaks preceding the first cell.

Returns *cell* or *range*

Examples

```
clrvbreak(a1)

clrvbreak(a1:b7)

clrvbreak(myrange)
```

DEFAULTS

Summary Sets the formatting of the specified cell or range to its defaults.

Usage DEFAULTS(*cell*) or DEFAULTS(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function changes the format of the selected *cell* or *range* to the defaults for its current style template. For more information about style templates, see "Style Template submenu" on page 155.

Returns *cell* or *range*

Examples

```
defaults(a1)

defaults(a1:b7)

defaults(myrange)
```

HBREAK

Summary Sets a horizontal page break.

Usage HBREAK(*cell*) or HBREAK(*range*)

Arguments
cell any cell reference
range any worksheet range

Description Sets a horizontal page break immediately after *cell* or *range* .

Returns *cell* or *range*

Examples

```
hbreak(a1)
hbreak(a1:b7)
hbreak(myrange)
```

HIDDENCOL

Summary Tests for hidden columns.

Usage HIDDENCOL(*cell*)

Arguments
cell any cell reference

Description This functions tests if *cell* is in a hidden column.

Returns 1 if *cell* is hidden, 0 if not

Examples

```
hiddencol(h19)
```

HIDDENROW

Summary Tests for hidden rows.

Usage HIDDENROW(*cell*)

Arguments *cell* any cell reference

Description This function tests if *cell* is in a hidden row .

Returns 1 if *cell* is hidden, 0 if not

Examples

```
hiddenrow(h19)
```

HIDECOL

Summary Hides the specified columns.

Usage HIDECOL(*cell*) or HIDECOL(*range*)

Arguments *cell* any cell reference
range any worksheet range

Description This function hides the columns that contain *cell* or *range* . Column A cannot be hidden.

Returns *cell* or *range*

Examples

```
hidecol(b5)
hidecol(b5:c7)
hidecol(myrange)
```

HIDEROW

Summary Hides the specified rows.

Usage HIDEROW(*cell*) or HIDEROW(*range*)

Arguments

cell any cell reference
range any worksheet range

Description This function hides the rows that contain *cell* or *range* . Row 1 cannot be hidden.

Returns *cell* or *range*

Examples

```
hiderow(b5)
hiderow(b5:c7)
hiderow(myrange)
```

ITALIC

Summary Toggles the data in the specified range between *italic* and standard text.

Arguments

cell any cell reference
range any worksheet range

Description This function changes the format of the cells specified by *cell* or *range* to *italic* if they are standard, and to standard if they are *italic* .

Returns *cell* or *range*

Examples

```
italic(a1)
italic(a1:b7)
italic(myrange)
```

OUTLINE

Summary Outlines the specified cells with the specified border .

Usage OUTLINE(*cell* , *border*) or OUTLINE(*range* , *border*)

Arguments

cell	any cell reference
range	any worksheet range
border	the specified border type: 0, 1 or 2

Description This function outlines each of the specified cells with a border of type *border* . Possible borders are 0 (no border), 1 (single border) or 2 (double border).

Returns *cell* or *range*

Examples

```
outline(a1,1)
outline(a1:b7,0)
outline(myrange,2)
```

PLAIN

Summary Changes the format of the specified cell or range to plain text.

Usage PLAIN(*cell*) or PLAIN(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function changes the text formatting of *cell* or *range* to plain (i.e. non- **bold** , non- *italic)* .

Returns *cell* or *range*

Examples

```
plain(a1)
plain(a1:b7)
plain(myrange)
```

RESETROWSIZE

Summary Resets the size of the specified rows.

Usage RESETROWSIZE(*cell*) or RESETROWSIZE(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function sets the row(s) that contain *cell* or *range* to the default for the worksheet.

Returns *cell* or *range*

Examples

```
resetrowsize(a1)
resetrowsize(a1:b7)
resetrowsize(myrange)
```

RESETCOLSIZE

Summary Resets the size of the specified columns.

Usage RESETCOLSIZE(*cell*) or RESETCOLSIZE(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description This function sets the column(s) that contain *cell* or *range* to the default for the worksheet.

Returns *cell* or *range*

Examples

```
resetcolsize(a1)

resetcolsize(a1:b7)

resetcolsize(myrange)
```

SETALIGN

Summary Sets the text alignment of the specified cell or range.

Usage SET ALIGN(*cell*, *h-align* [, *v-align*]) or
SET ALIGN(*range* , *h-align* [, *v-align*])

Arguments

cell	any cell reference
range	any worksheet range
h-align	a horizontal alignment constant
v-align	an [optional] vertical alignment constant

Description This function aligns the text in *cell* or *range* according to *h-align* and an optional *v-align* . For a list of alignment constants, see “Horizontal Alignment Constants” on page 391 and “V ertical Alignment Constants” on page 392.

Returns *cell* or *range*

Examples

```
setalign(a1,LEFTALIGN)

setalign(a1:b7,RIGHTALIGN,TOPALIGN)

setalign(myrange,RIGHTALIGN,BOTTOMALIGN)
```

SETALTCOLOR

Summary

Sets the alternate text color of the specified cell or range.

Usage

SET ALTCOLOR(*cell*, *color* [, *pattern*]),
SET ALTCOLOR(*range* , *color* [, *pattern*]),
SET ALTCOLOR(*cell*, *r*, *g*, *b* [, *pattern*]), or
SET ALTCOLOR(*range* , *r*, *g*, *b* [, *pattern*])

Arguments

cell	any cell reference
range	any worksheet range
color	a color constant
r	the amount of red
g	the amount of green
b	the amount of blue
pattern	a pattern number

Description

This function sets the alternate text color of *cell* or *range* to the specified *color* and optional *pattern* . The color can be specified by a color constant *color* or with red, green, and blue values (*r*, *g*, *b*) between 0 and 1. For a list of color constants, see “Color constants” on page 391. The pattern can be specified with a number between 0 and 87.

Returns

cell or *range*

Examples

```
setaltcolor(a1,BLACKCOLOR)

setaltcolor(a1:b7,0,0.5,0.5)

setaltcolor(myrange,MAGENTACOLOR,53)
```

SETALTPATTERN

Summary Sets the alternate text pattern of the specified cell or range.

Usage SET ALTPATTERN(*cell*, *pattern*) or
SET ALTPATTERN(*range* , *pattern*)

Arguments

cell	any cell reference
range	any worksheet range
pattern	a pattern number

Description This function sets the alternate text pattern of *cell* or *range* to *pattern* , using the current alternate color . The pattern can be specified with a number between 0 and 87.

Returns *cell* or *range*

Examples

```
setaltpattern(a1,23)

setaltpattern(a1:b7,17)

setaltpattern(myrange,59)
```

SETBKGCOLOR

Summary

Sets the background color of the specified cell or range.

Usage

SETBKGCOLOR(*cell*, *color* [, *pattern*]),
SETBKGCOLOR(*range* , *color* [, *pattern*]),
SETBKGCOLOR(*cell*, *r*, *g*, *b* [, *pattern*]), or
SETBKGCOLOR(*range* , *r*, *g*, *b* [, *pattern*])

Arguments

cell	any cell reference
range	any worksheet range
color	a color constant
r	the amount of red
g	the amount of green
b	the amount of blue
pattern	a pattern number

Description

This function sets the background color of *cell* or *range* to the specified *color* and optional *pattern* . The color can be specified by a color constant *color* or with red, green, and blue values (*r*, *g*, *b*) between 0 and 1. For a list of color constants, see “Color constants” on page 391. The pattern can be specified with a number between 0 and 87.

Returns

cell or *range*

Examples

```
setbkgcolor(a1,BLACKCOLOR)

setbkgcolor(a1:b7,0,0.5,0.5)

setbkgcolor(myrange,MAGENTACOLOR,53)
```

SETBKGPATTERN

Summary Sets the background pattern of the specified cell or range.

Usage SETBKGP ATTERN(*cell*, *pattern*) or
SETBKGP ATTERN(*range* , *pattern*)

Arguments

cell	any cell reference
range	any worksheet range
pattern	a pattern number

Description This function sets the background pattern of *cell* or *range* to *pattern* , using the current background color . The pattern can be specified with a number between 0 and 87.

Returns *cell* or *range*

Examples

```
setbkgpattern(a1,23)

setbkgpattern(a1:b7,17)

setbkgpattern(myrange,59)
```

SETBORDER

Summary Sets the border for a specified cell or range.

Usage SETBORDER(*range* , *borders* , *bool*)

Arguments

range	any worksheet range
borders	a border constant
bool	whether the border should be visible

Description This function places a border in *range* around the sides of the cells specified by *borders* , which can be one of: `LEFTBORDER`, `RIGHTBORDER`, `TOPBORDER`, `BOT-`

TOMBORDER. These constants can be OR'd together . The *bool* argument should be TRUE or F ALSE, depending on whether the border should be visible.

Returns *range*

Examples

```
setborder(a1,TOPBORDER,TRUE)

setborder(a1:b7,LEFTBORDER | RIGHTBORDER,TRUE)

setborder(myrange,TOPBORDER | BOTTOMBORDER,FALSE)
```

SETBORDERCOLOR

Summary Sets the border color for a specified cell or range.

Usage SETBORDERCOLOR(*cell*, *color* [, *pattern*]), SETBORDERCOLOR(*range* , *color* [, *pattern*]), SETBORDERCOLOR(*cell*, *r*, *g*, *b* [, *pattern*]), or SETBORDERCOLOR(*range* , *r*, *g*, *b* [, *pattern*])

Arguments

cell	any cell reference
range	any worksheet range
color	a color constant
r	the amount of red
g	the amount of green
b	the amount of blue
pattern	a pattern number

Description This function sets the border color of *cell* or *range* to the specified *color* and optional *pattern* . The color can be specified by a color constant *color* or with red, green, and blue values (*r*, *g*, *b*) between 0 and 1. For a

list of color constants, see “Color constants” on page 391. The pattern can be specified with a number between 0 and 87.

Returns *cell* or *range*

Examples

```
setbordercolor(a1,BLACKCOLOR)
setbordercolor(a1:b7,0,0.5,0.5)
setbordercolor(myrange,MAGENTACOLOR,53)
```

SETBORDERPATTERN

Summary Sets the border pattern for the specified cell or range.

Usage SETBORDERP ATTERN(*cell*, *pattern*) or
SETBORDERP ATTERN(*range* , *pattern*)

Arguments

cell	any cell reference
range	any worksheet range
pattern	a pattern number

Description This function sets the border pattern of *cell* or *range* to *pattern* , using the current border color . The pattern can be specified with a number between 0 and 87.

Returns *cell* or *range*

Examples

```
setborderpattern(a1,23)
setborderpattern(a1:b7,17)
setborderpattern(myrange,59)
```

SETBORDERTYPE

Summary Sets the border type for a specified cell or range.

Usage SETBORDERTYPE(*cell* , *border*) or
SETBORDERTYPE(*range* , *border*)

Arguments

cell	any cell reference
range	any worksheet range
border	a border type: 0, 1, or 2

Description This function changes the border for *cell* or *range* to border type *border* . Possible borders are 0 (no border), 1 (single border) or 2 (double border). This does not place a border around the cell.

Returns *cell* or *range*

Examples

```
setbordertype(a1,1)

setbordertype(a1:b7,0)

setbordertype(myrange,2)
```

SETCOLOR

Summary Sets the text color for a specified cell or range.

Usage

SETCOLOR(*cell*, *color* [, *pattern*]),
SETCOLOR(*range* , *color* [, *pattern*]),
SETCOLOR(*cell*, *r*, *g*, *b* [, *pattern*]), or
SETCOLOR(*cell*, *r*, *g*, *b* [, *pattern*])

Arguments

cell	any cell reference
range	any worksheet range
color	a color constant
r	the amount of red
g	the amount of green
b	the amount of blue
pattern	a pattern number

Description This function sets the text color of *cell* or *range* to the specified *color* and optional *pattern* . The color can be specified by a color constant *color* or with red, green, and blue values (*r*, *g*, *b*) between 0 and 1. For a list of color constants, see “Color constants” on page 391. The pattern can be specified with a number between 0 and 87.

Returns *cell* or *range*

Examples

```
setcolor(a1,BLACKCOLOR)

setcolor(a1:b7,0,0.5,0.5)

setcolor(myrange,MAGENTACOLOR,53)
```

SETCOLSIZE

Summary Sets the size of the specified columns.

Usage SETCOLSIZE(*cell*, *num*) or SETCOLSIZE(*range* , *num*)

Arguments

cell	any cell reference
range	any worksheet range
num	the size of the column(s) in points

Description This function sets the column(s) that contain *cell* or *range* to a size of *num* points.

Returns *cell* or *range*

Examples

```
setcolsize(a1,72)

setcolsize(a1:b7,100)

setcolsize(myrange,53)
```

SETFMT or SETDISPLAYFMT

Summary Sets the display format of the specified cell or range.

Usage SETFMT(*cell*, *format* [, *precision*]) or
SETFMT(*range* , *format* [, *precision*])

Arguments

cell	any cell reference
range	any worksheet range
format	the size of the column(s) in points
precision	the [optional] number of decimal places.

Description This function sets the display format of *cell* or *range* to the format specified by *format* and, optionally , to *precision* decimal places. For a list of formatting constants, see "Formatting Constants" on page 392.

Returns *cell* or *range*

Examples

```
setfmt(a1:b10,CURRENCYFMT,2)
```

SETFONT

Summary Sets the font of the specified cell or range.

Usage SETFONT(*cell*, *fontname* , *size*) or SETFONT(*range* , *fontname* , *size*)

Arguments

cell	any cell reference
range	any worksheet range
fontname	the name of a font
size	the point size

Description This function changes the font in *cell* or *range* to the font *fontname* and point size *size* .

Returns *cell* or *range* , or -1 on error

Examples

```
setfont(a1:b7,ZapfChancery-MediumItalic,12)
```

SETPATTERN

Summary Sets the text pattern for the specified cell or range.

Usage SETP ATTERN(*cell*, *pattern*) or SETP ATTERN(*range* , *pattern*)

Arguments

cell	any cell reference
range	any worksheet range
pattern	a pattern number

Description This function sets the text pattern of *cell* or *range* to *pattern* , using the current text color . The pattern can be specified with a number between 0 and 87.

Returns *cell* or *range*

Examples

```
setpattern(a1,23)

setpattern(a1:b7,17)

setpattern(myrange,59)
```

SETPRECISION

Summary Sets the precision of the specified cell or range.

Usage SETPRECISION(*cell*, *precision*) or
SETPRECISION(*range* , *precision*)

Arguments

cell	any cell reference
range	any worksheet range
precision	the number of decimal places.

Description Sets the number of decimal places for *cell* or *range* to *precision* .

Returns *cell* or *range*

Examples

```
setprecision(a1,2)

setprecision(a1:b7,3)

setprecision(myrange,0)
```

SETROWSIZE

Summary Sets the size of the specified rows.

Usage SETROWSIZE(*cell*, *num*) or SETROWSIZE(*range* , *num*)

Arguments

cell	any cell reference
range	any worksheet range
num	the size of the row(s) in points

Description This function sets the row(s) that contain *cell* or *range* to a size of *num* points.

Returns *cell* or *range*

Examples

```
setrowsize(a1,72)

setrowsize(a1:b7,100)

setrowsize(myrange,53)
```

SETTEMPLATE

Summary Sets the style template of the specified cell or range.

Usage SETTEMPLA TE(*cell* , *template*) or
SETTEMPLA TE(*range* , *template*)

Arguments

cell	any cell reference
range	any worksheet range
template	a style template number

Description Sets the style template for *cell* or *range* to *template* . *template* can be an integer between 1 and 8, inclusive. For more information about style templates, see “Style T emplate submenu” on page 155.

Returns *cell* or *range*

Examples

```
setrowsize(a1,1)

setrowsize(a1:b7,6)

setrowsize(myrange,3)
```

SETVERTALIGN

Summary

Sets the vertical text alignment of the specified cell or range.

Usage

SETVERT ALIGN(*cell*, *alignment*) or
SETVERT ALIGN(*range* , *alignment*)

Arguments

cell	any cell reference
range	any worksheet range
alignment	a vertical alignment constant

Description

This function vertically aligns the text in *cell* or *range* according to *alignment* . For a list of alignment constants, see “V ertical Alignment Constants” on page 392.

Returns

cell or *range*

Examples

```
setalign(a1,TOPALIGN)

setalign(a1:b7,TOPALIGN)

setalign(myrange,BOTTOMALIGN)
```

SMARTCOLSIZE

Summary Smartsizes the columns of the specified cell or range.

Usage SMARTCOLSIZE(*cell*) or SMARTCOLSIZE(*range*)

Arguments
cell any cell reference
range any worksheet range

Description This function sets the column(s) that contain *cell* or *range* to the best column size for the data they contain.

Returns *cell* or *range*

Examples

```
smartcolsize(a1)
smartcolsize(a1:b7)
smartcolsize(myrange)
```

SMARTROWSIZE

Summary Smartsizes the rows of the specified cell or range.

Usage SMARTROWSIZE(*cell*) or SMARTROWSIZE(*range*)

Arguments
cell any cell reference
range any worksheet range

Description This function sets the row(s) that contain *cell* or *range* to the best row size for the data they contain.

Returns *cell* or *range*

Examples

```
smartrowsize(a1)
smartrowsize(a1:b7)
smartrowsize(myrange)
```

UNHIDECOL

Summary Unhides the specified columns.

Usage UNHIDECOL(*cell*) or UNHIDECOL(*range*)

Arguments
cell any cell reference
range any worksheet range

Description This function unhides any hidden columns in *cell* or *range* .

Returns *cell* or *range*

Examples

```
unhidecol(a1)
unhidecol(a1:b7)
unhidecol(myrange)
```

UNHIDEROW

Summary Unhides the specified rows.

Usage UNHIDEROW(*cell*) or UNHIDEROW(*range*)

Description This function unhides any hidden rows in *cell* or *range* .

Returns *cell* or *range*

Examples

```
unhiderow(a1)
unhiderow(a1:b7)
unhiderow(myrange)
```

VBREAK

Summary Sets a vertical page break.

Usage VBREAK(*cell*) or VBREAK(*range*)

Arguments

cell	any cell reference
range	any worksheet range

Description Sets a vertical page break immediately after *cell* or *range* .

Returns *cell* or *range*

Examples

```
vbreak(a1)

vbreak(a1:b7)

vbreak(myrange)
```

Graphics Operations

ABOVESHADING

Summary Places a graphic above the shading on the sheet.

Usage ABOVESHADING(*name*)

Arguments *name* the name of a graphic

Description This function places the graphic named by *name* above the shading but below the text on the sheet. *name* must be a string.

Returns *name* , or -1 if *name* isn't found

Examples

```
aboveshading("Graph 1")

c1 := "MyGraphic"
aboveshading(c1)
```

ABOVESHEET

Summary Places a graphic about the sheet.

Usage ABOVESHEET(*name*)

Arguments *name* the name of a graphic

Description This function places the graphic named by *name* above the sheet. *name* must be a string.

Returns *name* , or -1 if *name* isn't found

Examples

```
abovesheet("Graph 1")

c1 := "MyGraphic"
abovesheet(c1)
```

BRINGTOFRONT

Summary Brings a graphic to the front of its sheet layer .

Usage BRINGTOFRONT(*name*)

Arguments *name* the name of a graphic

Description This function brings the graphic named by *name* to the front of other graphics in the same layer on the sheet. *name* must be a string.

Returns *name* , or -1 if *name* isn't found

Examples

```
bringtofront("Graph 1")

c1 := "MyGraphic"
bringtofront(c1)
```

CLEARGRAPH

Summary Clears a graph.

Usage CLEARGRAPH(*name*)

Arguments *name* the name of a graph

Description This function clears the graph named by *name* from the sheet. *name* must be a string.

Returns *name* , or -1 if *name* isn't found

Examples

```
cleargraph("Graph 1")

c1 := "MyGraphic"
cleargraph(c1)
```

GRAPH

Summary Creates a graph.

Usage GRAPH(*range* , *type* [, *name*])

Arguments

range	any worksheet range
type	a graph type constant
name	the [optional] graph name

Description This function graphs *range* in a graph of type *type* . For a list of graph type constants, see "Graph Constants" on page 394. Optionally , the graph can be given a name of *name* . *name* must be a string. If a *name* is not chosen, the name assigned is "Graph *n* ", where *n* starts at 1 and is incremented with each successive graph.

Returns *name* , or -1 if there's an error

Examples

```
graph(a1:c10,BARGRAPH)

graph(d10:f17,COLUMNGRAPH,"MyGraph")
```

SENDTOBACK

Summary Sends a graphic to the back of its sheet layer .

Usage SENDTOBACK(*name*)

Arguments *name* the name of a graphic

Description This function sends the graphic named by *name* to the back of other graphics in the same layer on the sheet. *name* must be a string.

Returns *name* , or -1 if *name* isn't found

Examples

```
sendtoback("Graph 1")

c1 := "MyGraphic"
sendtoback(c1)
```

SETGRAPHSIZE

Summary Changes the size of the specified graph.

Usage SETGRAPHSIZE(*name* , *width* , *height*)

Arguments

name the name of a graphic
width the width of the graph in points
height the height of the graph in points

Description This function sets the *width* and *height* of the graph *name* , in points.

Returns *name* , or -1 if *name* isn't found

Examples

```
setgraphsize("Graph 1",520,350)
```

SETGRAPHXY

Summary Sets the X and Y coordinates of a graph.

Usage SETGRAPHXY(*name* , *x*, *y*)

Arguments

name	the name of a graphic
x	the width of the graph in points
y	the height of the graph in points

Description This function sets the X and Y coordinates, in points, of the graph *name* to *x* and *y*, respectively . The X coordinate is measured from the left of the worksheet, and the Y coordinate is measured from the top of the worksheet. The name *name* must be a string.

Returns *name* , or -1 if *name* isn't found

Examples

```
setgraphxy("Graph 1",100,200)
```

UNDERALL

Summary Places a graphic under everything on a sheet.

Usage UNDERALL(*name*)

Arguments

name	the name of a graphic

Description This function places the graphic named by *name* under everything in the sheet, including text and shading. *name* must be a string.

Returns *name* , or -1 if *name* isn't found

Examples

```
underall("Graph 1")

c1 := "MyGraphic"
underall(c1)
```

Other Operations

ADDIN

Summary Calls an EventAddIn with optional data.

Usage ADDIN(*name* [, *data* ...])

Arguments

name the name of an EventAddIn
data the [optional] data to be passed to the AddIn

Description This function calls the EventAddIn *name* , optionally passing it the specified *data* . The name *name* must be expressed as a string.

Returns the return value from the AddIn

Examples

```
addin("MyAddIn")
```

ALERT

Summary Opens an alert panel.

Usage ALERT(*title* , *message* [, *button1* [, *button2* [, *button3*]]])

Arguments

title	the title of the alert panel
message	the message of the alert panel
button N	the titles of the buttons

Description This function opens an alert panel with the *title* , *message* , and *button* s specified. The *title, message* , and *button* N names must be strings. If no buttons are specified, one button will appear , labeled *OK*.

Returns the number of the button pressed, with the first button numbered 0

Examples

```
alert("Warning","The number entered exceeds the
maximum.", "Redo","Accept")
```

DISPLAY

Summary Re-displays the worksheet.

Usage DISPLA Y()

Arguments <none>

Description This function re-displays the worksheet. By default, the sheet is only re-displayed at the end of a script. This is useful for re-displaying the worksheet during the execution of a long script.

Returns 0

Examples

```
display()
```

FREEZE

Summary Freezes the value of a cell reference.

Usage FREEZE(*value*)

Arguments *value* any cell reference

Description This function returns the current value of *cell*. This is useful for assignment of the value in a cell to a local variable.

Returns the current value of

Examples

```
b1 := freeze(a1)
?x := freeze(g7)
```

INVERT

Summary Inverts a matrix.

Usage INVERT(*range* , *dest*)

Arguments *range* any worksheet range representing a matrix
dest the destination of the inverted matrix

Description This function inverts the matrix in *range* and places the inverse in *dest*. Both ranges must be square and of the same size.

Returns 0 if successful and -1 if unsuccessful

Examples

```
invert(a1:c3,d1:f3)
invert(mymatrix,g7:j10)
```

LET

Summary Assigns a value to a variable.

Usage LET(*variable* , *value*)

Arguments
variable a variable name
value a number , string, label, or cell reference

Description This function sets a *variable* to a *value* . The *variable* must be expressed as a string. V ariables can also be assigned with the := operator .

Returns *value*

Examples
```
LET("?x",10).
% is the same as
?x := 10
```

MENU

Summary Displays a menu of choices.

Usage MENU(*range* [, MENUBYROW|MENUBYCOL])

Description This function gives the user a menu of choices. For each column in *range* , the first row is the short name, the second row is the long description, and the third row is the string to return (this can be used for a subsequent “goto” or “gosub”). It takes an optional parameter to specify if the orientation of the menu's data is vertical or horizontal. The default is MENUBYCOL. For more information, see “Menus” on page 316.

Returns the return string, or -1 if the *Cancel* button is clicked

Examples
```
?foo := menu(a1:c3,MENUBYROW)
```

MULTIPLY

Summary Multiplies two matrices.

Usage MUL TIPL Y(*Xrange* , *YRange* , *dest*)

Arguments

XRange a worksheet range representing a matrix
YRange a worksheet range representing a matrix
dest the destination range

Description This function multiplies the two matrices specified by *XRange* and *YRange* and places the result in *dest* .

Returns 0 if successful and -1 if unsuccessful

Examples

```
multiply(A1:C3,E1:F3,H1:I3)
```

PRINT

Summary Prints a report or a part of the worksheet.

Usage PRINT([*range*]) or PRINT([*report*])

Description This function prints the *range* or the *report* specified. Report names must be expressed as strings. If nothing is specified, PRINT() will print all the information on the sheet.

Returns 1 if success, -1 if failure

Examples

```
print()
print("MyReport")
print(a1:g27)
```

PROMPT

Summary Prompts the user for input.

Usage PROMPT(*text*)

Arguments *text* the user prompt string

Description This function prompts the user for input with the string *text* . The message *text* must be expressed as a string.

Returns the user's input, as a string

Examples

```
a1 := prompt("Type in any word.")
a2 := value(prompt("Type in any number"))
```

QUERY

Summary Executes an SQL query .

Usage QUER Y(*name*)

Arguments *name* the query name

Description This function executes the pre-defined SQL query *name* . *name* must be expressed as a string. The query must first be defined in the Query Inspector . For more information, see "Query Inspector" on page 115. This MScript command is not available in Sunrise.

Returns 0

Examples

```
query("myquery")
query(c7)
```

RECALC

Summary Recalculates the worksheet.

Usage RECALC(), RECALC(*cell*) or RECALC(*range*)

Description This function recalculates the *range* or, if no range is specified, the whole worksheet.

Returns *cell* or *range*

Examples

```
recalc()
recalc(b7:z99)
```

REGRESSION

Summary Performs a linear regression on the specified ranges.

Usage REGRESSION(*Xrange* , *YRange* , *dest*)

Arguments

XRange a worksheet range representing a matrix
YRange a worksheet range representing a matrix
dest the destination range

Description Performs linear regression on the input ranges specified by *XRange* and *YRange* and places the result in the range *dest* .

Returns 0 if successful and -1 if unsuccessful

Examples

```
regression(A1:C3,D1:D3,F1:F3)
```

REPORT

Summary Prints a specified report.

Usage REPORT(*name*)

Arguments *name* the name of a report

Description This function prints the pre-defined report *name* . *name* must be expressed as a string. The report must first be defined in the Report Layout Inspector . For more information about reports, see “Report Layout Inspector” on page 102.

Returns 0

Examples

```
report("myreport")
report(c18)
```

SCENARIO

Summary Runs a scenario.

Usage SCENARIO(*name*)

Arguments *name* the scenario name

Description This function runs the pre-defined scenario *name* . *name* must be expressed as a string. The scenario must first be defined in the Scenario Inspector . For more information about scenarios, see “Scenario Inspector” on page 124.

Returns 0

Examples

```
scenario("myscenario")
scenaro(c9)
```

SETADDRESS

Summary Sets the cell address for the macro execution.

Usage SET ADDRESS(*cell*)

Description This function sets the address of the *cell* that the macro is executing in. This is useful for the SAME() command. The default is A1.

Returns <nothing>

Examples

```
setadress(g17)
```

SETINPUT

Summary Sets the input of a specified cell.

Usage SETINPUT(*cell*, *string*)

Arguments

cell any cell reference
string a string

Description This function sets the input of the cell *cell* to *string* . The *string* will be interpreted by the cell as if typed by the user .

Returns the evaluated *string*

Examples

```
setinput(c1,"=a1+b1")
```

SETLABEL

Summary Assigns a label name to a range of cells.

Usage SETLABEL(*label* , *range*)

Arguments

label	any string
range	any worksheet range

Description This function assigns *range* to the name *label* . *label* must be expressed as a string.

Returns 0

Examples

```
setlabel("myrange",a1:g13)
```

SOLVE

Summary Finds an input value such that a specified formula will give a desired result.

Usage SOLVE(*var* , *comp* , *value* [, *prec*])

Arguments

var	any cell reference
comp	any cell reference containing a formula
value	the desired result
prec	the [optional] digits of precision

Description This function tries to find the correct value of the cell *var* such that the formula in *comp* yields *value* (a number). It places the result in *var* . Optionally , the number of digits of precision can be specified with *prec* . The default is 6.

Returns 0 if successful, -1 if not

Examples

```
solve(A1,B1,2.5)
```

```
solve(A1,B1,2.5,8)
```

SORT

Summary Sorts a range.

Usage SORT(*range* , *titles* , SORTBYROW | SORTBYCOL, *key1* , *sorttype* [, *key2* , *sorttype* [, *key3* , *sorttype*]])

Arguments

range	any worksheet range
titles	a boolean stating if the range has titles
key N	the offset of the column to sort by
sorttype	a sort-type constant (see below)

Description This functions sorts the given *range* . If *titles* is true, the top row of the range is ignored. T o sort by rows, use the SORTBYROW constant; to sort by columns, use the SORTBYCOL constant. The *key* is the number of rows or columns that the key row or column is offset from the top or left side of the range. Use ASCENDINGSORT or DESCENDINGSORT to specify the order of the sort. *key2* and *key3* are optional.

Returns 0

Examples

```
sort(a1:b10,false,SORTBYROW,0,ASCENDINGSORT)
```

SYSTEM

Summary Sends a string to the shell.

Usage SYSTEM(*string*)

Arguments *string* any string that can be interpreted by the shell

Description This function sends the string *string* to the shell for execution.

Returns the return value of the shell command

Examples

```
a1 := system("whoami")
```

Retrieving information about the sheet

GETALIGN

Summary Returns the horizontal alignment of a cell.

Usage GET ALIGN(*cell*)

Arguments *cell* any cell reference

Description This function returns the horizontal alignment of *cell*. For a list of alignment constants, see “Horizontal Alignment Constants” on page 391.

Returns an alignment constant

Examples

```
getalign(d19)
```

GETALTCOLOR

Summary Returns the alternate text color of a cell.

Usage GET ALTCOLOR(*cell*)

Arguments *cell* any cell reference

Description This function returns the alternate text color of *cell*, which can be compared to color constants, used for setting the colors of other cells, etc. The number returned is an internal Mesa number . It should only be used to compare to color constants or as a parameter in another color . For a list of color constants, see "Constants" on page 391.

Returns a color constant

Examples

```
getaltcolor(g12)
```

GETALTPATTERN

Summary Returns the alternate text pattern of a cell.

Usage GET ALTPATTERN(*cell*)

Arguments *cell* any cell reference

Description This function returns the alternate text pattern number of the cell *cell*. Pattern numbers are integers between 0 and 87, inclusive.

Returns a pattern number

Examples

```
getaltpattern(d18)
```

GETBKGCOLOR

Summary Returns the background color of a cell.

Usage GETBKGCOLOR(*cell*)

Arguments *cell* any cell reference

Description This function returns the background color of *cell*, which can be compared to color constants, used for setting the colors of other cells, etc. The number returned is an internal Mesa number . It should only be used to compare to color constants or as a parameter in another color operation. For a list of color constants, see "Constants" on page 391.

Returns a color constant

Examples

```
getbkgcolor(f10)
```

GETBKGPATTERN

Summary Returns the background pattern of a cell.

Usage GETBKGP ATTERN(*cell*)

Arguments *cell* any cell reference

Description This function returns the background pattern number of the cell *cell*. Pattern numbers are integers between 0 and 87, inclusive.

Returns a pattern number

Examples

```
getbkgpattern(f10)
```

GETBORDERCOLOR

Summary Returns the border color of a cell.

Usage GETBORDERCOLOR(*cell*)

Arguments *cell* any cell reference

Description This function returns the border color of *cell*, which can be compared to color constants, used for setting the colors of other cells, etc. The number returned is an internal Mesa number . It should only be used to compare to color constants or as a parameter in another color operation. For a list of color constants, see "Constants" on page 391.

Returns a color constant

Examples

```
getbordercolor(f10)
```

GETBORDERPATTERN

Summary Returns the border pattern of a cell.

Usage GETBORDERP ATTERN(*cell*)

Arguments *cell* any cell reference

Description This function returns the border pattern number of the cell *cell*. Pattern numbers are integers between 0 and 87, inclusive.

Returns a pattern number

Examples

```
getborderpattern(f10)
```

GETBORDERTYPE

Summary Returns the border type of a cell.

Usage GETBORDERTYPE(*cell*)

Arguments *cell* any cell reference

Description Returns the border type of *cell*, which will be 0 (no border), 1 (single border) or 2 (double border).

Returns a border type constant

Examples

```
getbordertype(g19)
```

GETCOLOR

Summary Returns the text color of a cell.

Usage GETCOLOR(*cell*)

Arguments *cell* any cell reference

Description This function returns the color of *cell*, which can be compared to color constants, used for setting the colors of other cells, etc. The number returned is an internal Mesa number . It should only be used to compare to color constants or as a parameter in another color operation. For a list of color constants, see "Constants" on page 391.

Returns a color constant

Examples

```
getcolor(b7)
```

GETCOLSIZE

Summary Returns the column size of a cell.

Usage GETCOLSIZE(*cell*)

Arguments *cell* any cell reference

Description This function returns the column size of *cell*, in points.

Returns the column size in points

Examples

```
?x := getrowsize(j20)
```

GETFMT or GETDISPLAYFMT

Summary Returns the format of a cell.

Usage GETFMT(*cell*)

Arguments *cell* any cell reference

Description This function returns the display format of *cell*, which can be compared to the format constants or used to set the format of other cells. For a list of formatting constants, see “Formatting Constants” on page 392.

Returns a formatting constant

Examples

```
getfmt(g18)
```

GETGRAPHHEIGHT

Summary Returns the height of a graph.

Usage GETGRAPHHEIGHT(*name*)

Description Returns the height of the graph *name* , in points. The *name* must be expressed as a string.

Returns the height of *name* in points

Examples

```
getgraphheight("mygraph")
```

GETGRAPHWIDTH

Summary Returns the width of a graph.

Usage GETGRAPHWIDTH(*name*)

Description Returns the width of the graph *name* , in points. The *name* must be expressed as a string.

Returns the width of *name* in points

Examples

```
getgraphwidth("mygraph")
```

GETGRAPHX

Summary Returns the X coordinate of a graph.

Usage GETGRAPHX(*name*)

Arguments *name* a graph name

Description This function returns the X coordinate of the graph *name* , in points, measured from the left side of the worksheet. The name *name* must be expressed as a string.

Returns the X coordinate of the graph in points

Examples

```
getgraphx("mygraph")
```

GETGRAPHY

Summary Returns the Y coordinate of a graph.

Usage GETGRAPHY(*name*)

Arguments *name* a graph name

Description This function returns the Y coordinate of the graph *name* , in points, measured from the left side of the worksheet. The name *name* must be expressed as a string.

Returns the Y coordinate of the graph in points

Examples

```
getgraphy("mygraph")
```

GETINPUT

Summary Returns the input string of a cell.

Usage GETINPUT(*cell*)

Arguments *cell* any cell reference

Description This function returns the user -typed input string for the *cell*.

Returns the input string

Examples

```
getinput(b17)
```

GETPATTERN

Summary Returns the text pattern of a cell.

Usage GETP ATTERN(*cell*)

Arguments *cell* any cell reference

Description This function returns the text pattern number of the cell *cell*. Pattern numbers are integers between 0 and 87, inclusive.

Returns a pattern number

Examples

```
getpattern(b7)
```

GETPRECISION

Summary Returns the precision of a cell.

Usage GETPRECISION(*cell*)

Arguments *cell* any cell reference

Description Returns the precision (number of decimal places shown) of *cell*.

Returns an integer

Examples

```
a2 := getprecision(g3)
```

GETROWSIZE

Summary Returns the row size of a cell.

Usage GETROWSIZE(*cell*)

Arguments *cell* any cell reference

Description This function returns the row size of *cell*, in points.

Returns the row size in points

Examples

```
?x := getrowsize(j20)
```

GETTEMPLATE

Summary Returns the style template of a cell.

Usage GETTEMPLA TE(*cell*)

Arguments *cell* any cell reference

Description This function returns the number of the style template number for the cell *cell*. A style template number is an integer between 1 and 8.

Returns a style template number

Examples

```
?y := gettemplate(b9)
```

GETVERTALIGN

Summary Returns the vertical text alignment of a cell.

Usage GETVERT ALIGN(*cell*)

Arguments *cell* any cell reference

Description This function returns the vertical alignment of *cell*. For a list of vertical alignment constants, see “V erti-cal Alignment Constants” on page 392.

Returns a vertical alignment constant

Examples

```
setalign(b3,getvertalign(g43))
```

Constants

Color constants

Constant	Red	Green	Blue
BLACKCOLOR	0	0	0
BLUECOLOR	0	0	1
CYANCOLOR	0	1	1
DKGRAYCOLOR	0.333	0.333	0.333
GREENCOLOR	0	1	0
LTGRAYCOLOR	0.666	0.666	0.666
MAGENTACOLOR	1	0	1
WHITECOLOR	1	1	1
YELLOWCOLOR	1	1	0
REDCOLOR	1	0	0

Horizontal Alignment Constants

Constant	Effect
RIGHTALIGN	Aligns text to right side of cell
LEFTALIGN	Aligns text to left side of cell
CENTERALIGN	Centers text in cell
SMARTALIGN	Aligns strings left, numbers and dates right
FILLALIGN	Makes copies of cell contents to fill the cell

Vertical Alignment Constants

Constant	Effect
TOP ALIGN	Aligns text to top of cell
BOTTOMALIGN	Aligns text to bottom of cell
WRAP ALIGN	Wraps text to fit inside width of cell

Formatting Constants

Constant	Effect
GENERALFMT	Displays numbers as typed
FIXEDFMT	Displays numbers in fixed decimal format with the default number of decimal places
SCIENTIFICFMT	Displays numbers in scientific notation
CURRENCYFMT	Displays numbers in the default currency format
COMMAFMT	Displays numbers using commas to separate groups of three digits
CHARTFMT	Displays numbers as a series of plus signs of length equal to the number
PERCENTFMT	Displays numbers in percentage format with the default number of decimal places

Formatting Constants

Constant	Effect
TEXTFMT	Displays formulas directly , not as values
HIDDENFMT	Hides numbers
DMYFMT	Treats number as a serial number of a date and displays the date in day-month-year format
DMFMT	Displays the date in day-month format
MYFMT	Displays the date in month-year format
HMSFMT	Number must be fractional; treats number as a time and displays it in twelve-hour hours:minutes-second format
HMFMT	Displays the time in twelve-hour hours:minutes format
DATE1FMT	Displays date in month/day/year format
DATE2FMT	Displays date in month/day format
TIME1FMT	Displays time in 24-hour hour:minutes:seconds format
TIME2FMT	Displays time in 24-hour hour:minutes format

Graph Constants

Constant	Meaning
BARGRAPH	Bar chart
STACKEDBARGRAPH	Stacked bar chart
COLGRAPH	Column chart
STACKEDCOLGRAPH	Stacked column chart
SCATTERGRAPH	Scatter chart
XYGRAPH	X-Y chart
AREAGRAPH	Area chart
PIEGRAPH	Pie chart
HILOGRAPH	High/Low chart
HILOCLOSEGRAPH	Hi/Low/Close chart
BAR3DGRAPH	3D bar chart
RIBBONGRAPH	Ribbon chart
PYRAMIDGRAPH	Pyramid chart
SURF ACEGRAPH	Surface chart

Error Constants

Constant	Meaning
ERR	General error
ERRORCONST	General error
NA	Special NA error
NACONST	Special NA error

MesaBook.book : MScript.frame Black #91

9. Frequently Asked Questions

Here is a list of questions that are frequently asked about Mesa, along with their answers.

Negative currency values show up in parenthesis. Is there any other way to show negative currency?

Using parenthesis is the standard way of showing negative currency values. When using the currency format, negative values will always be shown in parenthesis.

Negative values are shown in the alternate color. Currently, the alternate color and the normal text color are both black by default. If you would like for negative values to show up in a different color, you can change the alternate color using the Cell Style Inspector or the Sheet Inspector.

In order to change the alternate color of the cells in the currency format, drag and drop the appropriate color from the color inspector to the indicator for the alternate color. This would in the color display of either the Cell Style Inspector of Sheet Inspector, depending on which one you used to change the format of the cells to currency.

Why won't my font change?

If you use the font panel to change your font, hitting the *Set* button will change the font in the highlighted cells only. If you want to change the font of the entire sheet, a graph, or a report, you need to open the appropriate inspector and drag and drop the font from the font panel to the font well in the inspector.

Example One: To change the font of the entire worksheet, open the Sheet Inspector and click on the *Font* button. The name of the default font will be displayed in a font window on the inspector. Clicking the *Fonts* button below the font window will open the Font Panel. Select the font you want and click the *Preview* button. *Command-drag* from the font well of the Font Panel to the font window on the Sheet Inspector. The new font name should now be displayed in the font window.

Example Two: To change the title of a graph from 12 point Helvetica Medium to 14 point Times Bold, open the Graph Inspector and click on the *Titles* button. Below each title entry window is a small font window that indicates what font the title is in. For example, "Helvetica 12 pt." If you wanted the first title of the graph to be in 14 point Times Bold, you would open the font panel, highlight "Times" under *Family*, "Bold" under *Typeface*, and "14" under *Size*. Then click on the *Preview* button and *Command-drag* from the font well to the font window under the first title. As you are doing this you should see a little font chip at the end of the cursor. The font window under the title should now say "Times-Bold 14." Note that the font in the entry window will not change, just the font of the title as it appears on the graph. You can change the font of each different title this way.

How do I get symbols such as gamma and sigma?

Mesa does not have any commands for Greek letter symbols. However, if you have a copy of the Symbol font on your machine, you may want to use that.

How do I get my graphs to show up in different colors and patterns?

Making graphs appear in different colors and patterns can be done easily by dragging color chips from the Colors Panel and pattern chips from the Patterns Panel, and dropping them onto the graph. For example, if you have a 3d Bar chart, and you drop a red color chip onto one of the bars, all the bars from that group of data will become red. Patterns work the same way. One thing to note about mixing colors and patterns is that the portion of the pattern that shows up as black in the pattern panel will be the part in the color that you chose.

The colors on Mesa's graphs follow their orientation. In graphs with a horizontal orientation, the data points within each column will have the same color; for graphs with a vertical orientation, the data points within each row will be graphed in the same color.

Why don't the buttons on the Inspector light up?

Many people are confused by the fact that the buttons such on inspectors such as the Query Inspector and Scenario Inspector are not always enabled. In these inspectors, the *Add* button remains disabled until all the relevant information, including a name, has been entered. The other buttons will become enabled once a query or scenario has been selected from the list of names.

How do I save a spreadsheet or graph as a PostScript file?

If you would like to save something as a PostScript file, select the **Print...** option from the main menu. Click on the *Save* button instead of the *Print* or *Preview* buttons. This will save the graph or worksheet as a PostScript file.

How do I get rid of blank pages in a report?

Blank pages in reports are often the result of blank cells being included in the report ranges. Check your ranges to make certain that they are correct. If you can not find the problem, try choosing the "Show Row and Column Titles" option in the Report Layout Inspector, and previewing the report. This will show you what Mesa is trying to print, and will often help solve the problem.

How do you get dates to show up as such with Imported Delimited T ext?

If you import dates in the form of @date(...) they will show up as @date(...) until you click on them, at which point the expression will evaluate. If you have imported dates from delimited text you must click on the cells containing dates if you what Mesa to evaluate them as dates. The reason for this is that delimited text is treated as simply a number of strings separated by delimiters.

What Lotus files does Mesa read?

Mesa reads .wks and .wk1 files, but *does not* read .wk3 files. If you need to read .wk3 files, use your Lotus program to save the file as a .wk1 file and then read it into Mesa.

What happens to the Lotus graphics when they are imported?

Mesa does not import Lotus graphs, so any graphs or charts you have in your Lotus files will not show up in Mesa. It should be fairly easy to recreate them using Mesa's Graph Inspector.

MesaBook.book : Questions.frame Black #6

10. Keyboard Reference

The following table contains a list of all Mesa keyboard commands, the menu name of the command, a brief description, and a cross reference to more information.

Keyboard Commands

Key Combination	Menu Name	Description	More Information
Command-?	Help	Start interactive Help	"Help" on page 137
Command-/	Slash Help	Convert 1-2-3 slash commands into Mesa commands	"Slash Help" on page 138
Command-!	Legal	Legal Information	"Legal" on page 138
Command-o	Open	Open a File	"Open" on page 139
Command-n	New	Create a New File	"New" on page 139
Command-s	Save	Save the Current File	"Save" on page 139
Command-Shift-S	Save As	Save Current File With New Name	"Save As..." on page 140
Command-u	Revert	Replace Current File W ith Last Saved	"Revert to Saved" on page 140
Command-Shift-W	Close	Close the Current Worksheet	"Close" on page 141

Keyboard Commands

Key Combination	Menu Name	Description	More Information
Command-x	Cut	Cut Current Selection to Pasteboard	"Cut" on page 143
Command-c	Copy	Copy Current Selection to Pasteboard	"Copy" on page 143
Command-v	Paste	Paste the Pasteboard to Current Selection	"Paste" on page 144
Command-z	Undo	Undo T yping	"Undo" on page 146
Command-Shift-Z	Redo	Redo T yping	"Redo" on page 146
Command-Shift-X	Blank Cell	Blank the Current Cell	"Blank Cell" on page 146
Command-a	Select All	Select the Entire Worksheet	"Select All" on page 146
Command-j	Insert	Insert Row/Column	"Insert" on page 148
Command-k	Delete	Delete Row/Column	"Delete" on page 149
Command-Shift-F	Formula Builder	Build a Formula	"Formula Builder" on page 149
Command-Shift-C	Cell Style Inspector	Bring up the Cell Style Inspector	"Cell Style Inspector" on page 150
Command-Shift-G	Sheet Inspector	Bring up the Sheet Inspector	"Sheet Inspector" on page 83
Command-b	Bold	Make Current Selection's Font Bold	"Cell Format submenu" on page 151
Command-i	Italic	Make Current Selection's Font Italic	"Cell Format submenu" on page 151
Command-Shift->	Bigger	Make Current Selection's Font Bigger	"Cell Format submenu" on page 151

Keyboard Commands

Key Combination	Menu Name	Description	More Information
Command-Shift- <	Smaller	Make Current Selection's Font Smaller	"Cell Format submenu" on page 151
Command-r	Right	Right Justify Current Selection	"Cell Format submenu" on page 151
Command-l	Left	Left Justify Current Selection	"Cell Format submenu" on page 151
Command-Shift- $	Currency	Display Current Selection as Currency	"Display Format submenu" on page 153
Command-Shift- %	Percent	Display Current Selection as Percentile	"Display Format submenu" on page 153
Command-,	Comma	Display Current Selection with Commas	"Display Format submenu" on page 153
Command-;	General	Display Current Selection as General	"Display Format submenu" on page 153
Command-Shift- @	Hour -Minute	Display Current Selection as Hour:-Minute	"Display Format submenu" on page 153
Command-Shift- #	Day-Month-Year	Display Current Selection as Day/Month/Y ear	"Display Format submenu" on page 153
Command-Shift- ^	Scientific	Display Current Selection in Scientific Notation	"Display Format submenu" on page 153
Command-Shift- _	No Border	Remove Border from Selected Cells	"Border submenu" on page 153

Keyboard Commands

Key Combination	Menu Name	Description	More Information
Command-1…8	Template 1…8	Change Current Selection to Style Template 1…8	“Style T emplate submenu” on page 155
Command-Shift-R	Report Layout	Open Report Layout Inspector	“Report Layout...” on page 155
Command-Shift-P	Page Layout	Open Page Layout Inspector	“Page Layout...” on page 156
Command-Shift-T	ToolBar	Display the T oolBar	“ToolBar” on page 158
Command-t	Fonts	Open the Font Panel	“Fonts” on page 157
Command-=	Recalculate	Recalculate W ork-sheet	“Recalculate” on page 159
Command-g	Go To Cell	Go To a Specified Cell or Range	“Goto Cell” on page 160
Command-Shift-M	Run Macro	Run a Macro from current cell	“Run Macro” on page 161
Command-Shift-&	Smart Sum	Sum the Current Selection	“Smart submenu” on page 162
Command-Shift-L	Label Inspector	Bring up the Label Inspector	“Labels submenu” on page 163
Command-f	Find	Find and Item in the Database	“Database submenu” on page 164
Command-Shift-I	Query Inspector…	Open the Query Inspector	“Query Inspector” on page 115
Command-Shift-N	New W indow	Create a New W indow From the Current W orksheet	“New W indow” on page 166

Keyboard Commands

Key Combination	Menu Name	Description	More Information
Command--	Send to Back	Send Selected Graphic to Back	"Send to Back" on page 182
Command-Shift-+	Bring to Front	Bring Selected Graphic to Front	"Bring to Front" on page 182
Command-m	Miniaturize	Miniaturize the Current W indow	"Miniaturize W indow" on page 184
Command-w	Close	Close the Current Window	"Close W indow" on page 184
Command-Shift-E	Close Error Panel	Close the Error Panel	"Close Error Panel" on page 185
Command-p	Print	Print the Current Sheet or Selection	"Print" on page 134
Command-h	Hide	Hide the Application	"Hide" on page 135
Command-q	Quit	Quit the Application	"Quit" on page 136
Shift- ↵	Fill Smart	Perform a SmartFill	
Alternate- ↵	Fill with Cell	Fill range with current cell	"Fill submenu" on page 161
Alternate- ← →	N/A	Move Cursor within Editing field	
Shift- ←∈ → ↑↓	N/A	Next non-empty cell	

MesaBook.book : Keyboard.frame Black #6

Symbols
!= 312
#NOT# 251
< 312
<= 312
<> 312
= 312
=< 312
== 312
=> 312
> 312
>= 312
@ @252
@ABS 198
@ACOS 198
@ADDRESS 252
@ASIN 199
@ATAN 200, 201
@AVE 203
@AVG 203
@BEEPIF 253
@BLACKCOLOR 301
@BLUECOLOR 301
@CALCRATE 225
@CELL 254
@CELLPOINTER 255
@CHAR 240
@CHARTFORMAT 302
@CHOOSE 256
@CLEAR 240
@CODE 241
@COLS 256
@COMMAFORMAT 302
@CONVDATE 277
@COS 204
@COUNT 205
@CTERM 226
@CURRENCYFORMAT 302
@CURRENTNUMBER 304
@CURRENTVALUE 304
@CYANCOLOR 301
@DATE 277
@DATEVALUE 278
@DAVE 287
@DAVG 287
@DAY 279
@DAYMONFORMAT 302
@DCOUNT 288
@DDB 227
@DEGTORAD 206
@DIDCHANGE 257
@DINTL1FORMAT 303
@DINTL2FORMAT 303
@DIV 207, 217
@DKGRAYCOLOR 301
@DMAX 290
@DMIN 292
@DMYFORMAT 302
@DPROD 294
@DSTD 297
@DSUM 298
@DVAR 300
@ERR 304
@EXACT 241
@EXP 208
@FALSE 304
@FILENAME 304
@FIND 242
@FIXEDFORMAT 301
@FORMAT 257
@FRAC 208
@GENERALFORMAT 301
@GREENCOLOR 301
@HISTOGRAM 258
@HLOOKUP 259
@HMSFORMAT 302

@HOSTNAME 304
@HOUR 279
@HOURMINFORMAT 302
@IF 260
@INDEX 263
@INT 209
@IRR 230
@ISEMPTY 264
@ISERR 264
@ISFORMULA 265
@ISNA 265
@ISNUMBER 266
@ISREF 266
@ISSTRING 267
@LEFT 243
@LEN 244
@LENGTH 244
@LN 209
@LOG 210
@LOWER 244
@LTGRAYCOLOR 301
@MAGENTACOLOR 301
@MAX 211
@MID 245
@MIN 213
@MINUTE 280
@MOD 214
@MONTH 280
@MONYEARFORMAT 302
@N 267
@NA 304
@NETWORKDAYS 281
@NEXT 58, 245, 268
@NEXTMONTH 281
@NEXTYEAR 282
@NOT 268
@NOW 303
@NPV 232
@PATHNAME 305
@PERCENTFORMAT 302
@PI 303
@PMT 234
@PROD 215
@PROPER 246
@PV 235
@RADTODEG 216
@RAND 303
@RANGE 269
@RATE 236
@REDCOLOR 301
@REPEAT 247
@REPLACE 247
@RIGHT 248
@ROOT 217
@ROUND 217
@ROWS 270
@S 270
@SAME 58, 271
@SCIENTIFICFORMAT 302
@SECOND 282
@SETALTCOLOR 272
@SETBKGCOLOR 273
@SETCOLOR 274
@SGN 218
@SIGN 218
@SIGNAL 275
@SIN 218
@SLN 237
@SQRT 219
@STD 221
@STRING 243
@SUM 222
@SYD 238
@TAN 222
@TERM 239
@TIME 283

@TIMEVALUE 283
@TINTL1FORMAT 303
@TINTL2FORMAT 303
@TODAY 303
@TRIM 248
@TRUE 305
@UPPER 249
@USERNAME 305
@VALUE 250
@VAR 224
@VLOOKUP 276
@WEEKDAY 284
@WHITECOLOR 301
@WORKDAY 284
@YEAR 285
@YELLOWCOLOR 301
^^ 312
|| 312

Numerics
1-2-3 3
3D Area Chart Command 180
3D Bar Chart 14
3D Bar Chart Command 177
3D graphs 100, 177, 178, 179, 180

A
About Mesa 2, 3
About Mesa command 137
ABOVESHADING 364
ABOVESHEET 364
ABS 198
ACOS 198
Add command 145
AddIn functions 73
AddIn Menu 185
ADDRESS 252, 314
address, setting 377
advanced topics
  global style settings 43
  Mesa preference settings 43
  tips and shortcuts 53
  tutorial 42
  worksheet features 44
ALERT 370
alignment 79, 151
alignment, changing 344, 360
alignment, getting 380, 390
alternate color, changing 272, 345
alternate color, getting 381
alternate patterns, changing 346
alternate patterns, getting 381, 382
AND 312
  logical 312
Area Chart command 175
Arguments 188
Arrange in Front command 183
ASCII 73, 137
ASIN 199
assigning variables 310, 372
ATAN 200
Athena Design 138
AutoGrow 60, 67, 68
AutoGrow Ranges 67
auto-grow ranges 55
automatic and manual recalculation 45
autosum option 55
AVE 202
AVERAGE 202

B
background color, changing 86, 152, 273, 347
background colors, getting 382
background patterns, changing 348
backgrounds, clear 86
back-solving 378

backup files 75
Bar Chart command 168
bar charts 97
base styles 58
basic topics - tutorial 17
BEEPIF 253
Bigger Font command 151
BLACKCOLOR 301, 391
Blank Cell command 146
BLUECOLOR 301, 391
BOLD 334
Bold command 151
border
    getting color 383
    getting pattern 383
    getting type 384
border color, getting 383
Border menu 153
border pattern, getting 383
border type, getting 384
borders 82, 109
    outlining 341
    setting 348
    setting color 349
    setting pattern 350
Branching 313
Bring to Front command 182
BRINGTOFRONT 365

C

CALCRATE 225
CELL 254
cell address, setting 377
Cell Addresses 189
cell color, changing 152, 274
Cell Colors menu 152
cell format 28
Cell Format menu 151
cell input, gettin 258, 388
cell input, setting 377
cell references, assigning to variables 310
Cell Style Inspector 62, 66, 75
Cell Style Inspector command 150
CELLPOINTER 255
Center command 152
changing column width 36
changing row height 37
CHAR 240
CHARTFORMAT 302
charts 366
CHOOSE 256
circular references 45
CLEAN 240
CLEAR 320
clear backgrounds 86
Clear command 147
Clear Special menu 147
CLEARDATA 321
CLEARFMT 321
CLEARFORMAT 321
CLEARGRAPH 365
CLEARHBREAK 335
clearing data 33
CLEARNUM 322
CLEARSTR 323
CLEARVBREAK 336
Close command 141
Close Error Panel command 185
Close Window command 184
CLR 320
CLRDATA 321
CLRFMT 321
CLRHBREAK 335
CLRNUM 322
CLRSTR 323
CLRVBREAK 336

CODE 241
color 78
Color Constants 301
color constants
BLACKCOLOR 391
BLUECOLOR 391
CYANCOLOR 391
DKGRAYCOLOR 391
GREENCOLOR 391
LTGRAYCOLOR 391
MAGENTACOLOR 391
REDCOLOR 391
WHITECOLOR 391
YELLOWCOLOR 391
color, changing 347
color, getting 381
Colors command 157
colors, changing 152, 272, 273, 274, 345, 352
colors, getting 382, 384
COLS 256
Column Chart 14
Column Chart command 170
column charts 97
column size, changing 343, 353, 361
column size, getting 385
column width, changing 36, 87
columns, hidden 338
columns, hiding 339, 362
Comma command 153
COMMAFORMAT 302
command key equivalents 19, 133
commands
3D Area Chart 180
3D Bar Chart 177
about Mesa 137
add 145
area chart 175
arrange in front 183
bar chart 168
bigger font 151
blank cell 146
bold 151
bring to front 182
cell style inspector 150
center 152
clear 147
close 141
close error panel 185
close window 184
colors 157
column chart 170
comma 153
copy 143
currency 153
cut 143
data 147
day-month-year 153
delete 149
delete all labels 164
direct input 183
divide 146
double border 153
event inspector 161
export tabular text 141
extract inspector 165
feedback 138
fill 152
fill down 161
fill right 161
fill smart 162
fill with cell 161
find 165
fixed 153
fonts 157
format 147
formula bar 183
formula builder 149
formulas to values 148
general 153
go to cell 160
graph inspector 181

help 137
hide 135
hide columns 154, 155
hide rows 154
high/low chart 176
high/low/close chart 177
hour-minute 153
import tabular text 141
insert 148
insert label table 164
italic 151
label inspector 163
label range to bottom 164
label range to left 164
label range to right 164
label range to top 164
layer inspector 182
left 152
legal 138
miniaturize window 184
move 144
mulitply 146
new 139
new window 166, 184
no border 153
numeric input 147
open 139
page layout 156
paste 144
paste format 144
paste values 144
patterns 157
percent 153
pie chart 173
plain 151
preferences 137
print 134
Pyramid Chart 179
query inspector 165
quit 136
recalc all 160
recalc range 160
recalculate 159
redo 146
report layout 155
reset column size 154
reset row size 154
revert to saved 140
Ribbon Chart 178
right 152
run macro 161
save 139
save as 140
scatter chart 171
scientific 153
select all 146
send to back 182
set current window tools as default 183
single border 153
slash help 138
smaller font 151
smart 152
smart column size 154
smart count 163
smart max 162
smart min 162
smart row size 154
smart std dev 163
smart variance 163
smartsum 162
sort 164
stacked bar chart 169
stacked column chart 170
string input 147
style template 155
subtract 145
to lower case 148
to proper case 148
to upper case 148
toolbar 158
transpose 145
undo 146
unhide rows 155

wrap 152
XY chart 172
Comments 308
Concurrence 4
conditional operators 312
constants
@CURRENTNUMBER 304
@CURRENTVALUE 304
@ERR 304
@FALSE 304
@FILENAME 304
@NA 304
@PATHNAME 305
@TRUE 305
BLACKCOLOR 391
BLUECOLOR 391
CURRENTNUMBER 304
CURRENTVALUE 304
CYANCOLOR 391
DKGRAYCOLOR 391
ERRORCONST 304, 395
FALSE 304
FILENAME 304
GREENCOLOR 391
LTGRAYCOLOR 391
MAGENTACOLOR 391
NACONST 305, 395
PATHNAME 305
REDCOLOR 391
THISCOLUMN 305
THISROW 305
TRUE 305
WHITECOLOR 391
YELLOWCOLOR 391
Constants (non-@) 301, 391
constrained ranges 56
control panel, learning 25
CONVDATE 277
Convert menu 148
converting
formulas to values 327
numbers to strings 329
strings to numbers 332
to lower case 328
to proper case 330
to upper case 333
converting formulas to values 50
COPY 324
Copy command 143
copy format 74
COPYFMT 325
COPYFORMAT 325
copying cell data 46
COPYVAL 326
COS 204
COUNT 204
count, smart 163
Create 4
Creating the Worksheet 60
Criteria Ranges 128
Currency command 153
Currency Symbol 74
CURRENCYFORMAT 302
CURRENTNUMBER 304
CURRENTVALUE 304
customizing the “home” key 53
Cut command 143
CYANCOLOR 301, 391

D

Data command 147
data entry options 183
data tables 124, 129
data, clearing 321
data, formatting 35
database
criteria ranges 128
extracts 128, 165
query 165
sort 164

Database Functions (non-@) 285
Database menu 164
DataPhile 4
DATE 277
Date Constants 303
Date Functions (non-@) 277
date serial number 189
Dates 189
DATEVALUE 278
DAVE 286
DAVERAGE 286
DAY 279
DAYMONFORMAT 302
Day-Month-Year command 153
DCOUNT 288
DDB 227
decimal place,getting 389
decimal places, setting 357
DEFAULTS 337
degrees, converting to radians 206
DEGTORAD 206
Delete All Labels command 164
Delete command 149
dialog panel 132
DIDCHANGE 257
dimensions 84
DINTL1FORMAT 303
DINTL2FORMAT 303
Direct Input 67
Direct Input command 183
DISPLAY 370
display format
getting 385
Display Format menu 153
display format, changing 354
display formats 35
display styles, worksheet 58
displaying a range table 49
DIV 207
Divide command 146
DKGRAYCOLOR 301, 391
DMAX 289
DMIN 291
DMYFORMAT 302
Double Border command 153
Double Input Scenarios 127
DPROD 293
Drag & Drop Ranges 11
DSTDDEV 296
DSTDEV 296
DSUM 298
DVAR 299

E
Edit menu 142
Editing a Script 306
editing data 27
e-mail message 138
Entering the Column Headings 62
Entering the Formulas 64
EPS 73, 137
ERR 304, 395
ERRORCONST 304, 395
Errors 308
Event Inspector 129
Event Inspector command 161
EXACT 241
Excel 3
exclusive OR 312
EXP 208
Export Tabular Text command 141
Export Worksheet 141
Extract Inspector 128
Extract Inspector command 165

F
FALSE 304
fax output 134

Feedback command 138
File menu 139
FILENAME 304
files
  importing 50
  linking 51
Fill command 152
Fill Down command 161
Fill menu 161
Fill Right command 161
Fill Smart command 162
Fill with Cell command 161
Financial Functions (non-@) 225
FIND 242
Find command 165
FIXED 243
Fixed command 153
FIXEDFORMAT 301
font chip 113, 158
font panel 157
fonts 81
  bold 334
  italic 340
  plain 342
  setting 355
Fonts command 157
footers 41, 110
FORMAT 257
format
  getting 385
Format command 147
Format Constants 301
Format menu 150
format, changing 354
format, clearing 321
format, copying 325
Formatting 65
formatting
  data 35
  reports 41
FORMTOVAL 327
Formula Bar 27, 68
Formula Bar command 183
Formula Builder 187
Formula Builder command 149
Formula Inheritance 12
formula to value conversion 50
Formulas 187
formulas 32
Formulas to Values command 148
FORMULASTOVAL 327
FRAC 208
FREEZE 371
freezing cell values 310
Function AddIns 73
Function Arguments 188
Functions (non-@) 186
FV 228

G
General command 153
GENERALFORMAT 301
GETALIGN 380
GETALTCOLOR 381
GETALTPATTERN 381
GETBKGCOLOR 382
GETBKGPATTERN 382
GETBORDERCOLOR 383
GETBORDERPATTERN 383
GETBORDERTYPE 384
GETCOLOR 384
GETCOLSIZE 385
GETDISPLAYFMT 385
GETFMT 385
GETGRAPHHEIGHT 386
GETGRAPHWIDTH 386
GETGRAPHX 387
GETGRAPHY 387

GETINPUT 258, 388
GETPATTERN 388
GETPRECISION 389
GETROWSIZE 389
GETTEMPLATE 390
Getting started 17
GETVERTALIGN 390
Go to Cell command 160
GRAPH 366
Graph 13
graph
  getting size 386
Graph Inspector 13, 14, 15, 88
Graph Inspector command 181
graphics
  bring to front 365
  cleargraph 365
  layers 364, 368
  send to back 367
  setting location 368
  setting size 367
Graphics menu 167
Graphing 13
graphs
  3D 100
  3D Area 180
  3D bar 177
  area 175
  bar 97
  bar chart 168
  changing background color 90
  column 97
  column chart 170
  getting location 387
  getting size 386
  grid 95
  high/low 176
  high/low/close 100, 177
  labels 90
  legends 96
  naming 89
  pie 173
  Pyramid 179
  ranges 96
  ribbon 178
  scale 93
  scatter 98, 171
  stacked bar chart 169
  stacked column 170
  titles 92
  XY 99, 172
GREENCOLOR 301, 391
grid
  graph 95
  worksheet 86

H

HBREAK 338
headers 41, 110
height of rows, changing 37
Help command 137
HIDDENCOL 338
HIDDENROW 339
Hide Columns command 154
Hide command 135
Hide Rows command 154
HIDECOL 339
HIDEROW 340
High/Low Chart command 176
High/Low/Close Chart command 177
Hiigh/Low/Close Charts 100
HISTOGRAM 258
HLOOKUP 259
HMSFORMAT 302
home key, customizing 53
HOSTNAME 304
HOUR 279
HOURMINFORMAT 302
Hour-Minute command 153
How to Use this Book 2

I
IF 260, 261
IFELSE 261
Import Tabular Text command 141
importing files 50
in-cell editing 53
INDEX 262
Info menu 137
Input Type 66
input type 83
input, getting from cell 258, 388
input, getting from user 374
input, setting for cell 377
inputting data 27, 183
Insert command 148
Insert Label Table command 164
Inspectors 24, 165
cell style inspector 75, 150
event inspector 129, 161
extract inspector 128, 165
graph inspector 88, 181
label inspector 114, 163
layer inspector 101, 182
MScript 161
pattern picker 131
preferences 72
query inspector 115, 165
report layout inspector 102
scenario 124, 129
sheet inspector 83
INT 209
Introduction 1
INVERT 371
IRR 229
ISEMPTY 264
ISERROR 264
ISFORMULA 265
ISNA 265
ISNUMBER 266
ISREF 266
ISSTRING 267
ITALIC 340
Italic command 151

K
Key Attributes 75
Keyboard 19
keys, special 2

L
Label Inspector 68, 114
Label Inspector command 163
Label Range menu 164
Label Range To Bottom command 164
Label Range To Left command 164
Label Range To Right command 164
Label Range To Top command 164
labelling a range 49
Labels 313
Labels menu 163
labels on graphs 90
landscape 113
Layer Inspector 101, 182
layering graphics 364, 368
learning the control panel 25
LEFT 243
Left command 152
Legal command 138
legends for graphs 96
LEN 244
LENGTH 244
LET 310, 372
License 74
Lighthouse Design 4
linear regression 375
linking files 51
LN 209

LOG 210
LOG10 210
Looping 313
Lotus 3
LOWER 244, 328
LTGRAYCOLOR 301, 391

M
macros 37, 73
macros, writing 161
MAGENTACOLOR 301, 391
Main menu 133
margins in reports 107
Math Constants 303
Math Functions (non-@) 198
matrices
  inverting 371
  multiplication 373
  regression 375
Matrix submenu 165
MAX 210
max, smart 162
MENU 372
Menus 19
menus
  addin menu 185
  border 153
  cell colors 152
  cell format 151
  clear special 147
  convert 148
  database 164
  display format 153
  edit 142
  file 139
  fill 161
  format 150
  graphics 167
  info 137
  keyboard shortcuts 133
  label range 164
  labels 163
  main 133
  page breaks 156
  paste special 144
  row and column size 154
  sheet 159
  smart 162
  style template 155
  windows 182
Mesa
  display formats 35
  license agreement 138
  license number 137
  technical support 17
  version number 137
MID 245
MIN 212
min, smart 162
Miniaturize Window command 184
MINUTE 280
MOD 214
MONTH 280
MONYEARFORMAT 302
MOVE 328
Move command 144
moving cell data 47
MScript Inspector 161
MULTIPLY 373
Multiply command 146

N
NA 304, 395
NACONST 305, 395
name of a chart 89
NAMESHEET 318
NETWORKDAYS 281
New command 139
New Window command 166, 184
NEXT 245, 267, 268

NeXT 2
Colors panel 157
computer 2
File Viewer 139, 141
Font Panel 157
pasteboard 143
Previewer 134
User's Reference 142
NEXT() 58
NEXTMONTH 281
NeXTstation 2
NEXTSTEP 17
NEXTYEAR 282
No Border command 153
non-scrolling titles 44
NOT 268
NOW 303
NPV 231
numeric data 32
Numeric Input command 147
numerical input, clearing 322
NUMTOSTR 329

O
Open command 139
Opening Non-Mesa Spreadsheets 18
Operators 192
operators
!= 312
< 312
<= 312
<> 312
= 312
=< 312
== 312
=> 312
> 312
>= 312
^^ 312
|| 312
exclusive OR 312
logical AND 312
logical OR 312
operators table 312
OR 312
exclusive 312
logical 312
OUTLINE 341

P
page breaks 335, 336, 338, 363
Page Breaks menu 156
Page Layout command 156
page size for reports 109
panels, alert 370
password 140
Paste command 144
Paste Format command 144
Paste Special menu 144
Paste Values command 144
pasteboard 143
PATHNAME 305
Pattern Picker 131
Patterns command 157
patterns, changing 346, 348, 356
patterns, getting 381, 382, 388
Percent command 153
PERCENTFORMAT 302
PI 303
Pie Chart command 173
PLAIN 342
Plain command 151
PMT 233
portrait 113
precision, getting 389
precision, setting 357
Preferences command 137
Preferences Panel 72
PRINT 373

Print command 134
print dialog panel 134
print topics - tutorial 38
printing 134
  default print settings 40
  defining a range to print 38
  landscape or portrait 113
  reports 114
printing reports 373, 376
PROD 215
PROMPT 374
PROPER 246, 330
protecting ranges 50
protection 86
PV 234
Pyramid Chart Command 179

Q
QUERY 374
Query Inspector 115
Query Inspector command 165
Quit command 136

R
radians, converting to degrees 216
RADTODEG 216
RAND 303
RANDOM 303
RANGE 269
range description formats 35
range table, displaying 49
ranges 22
  auto-grow 55
  constrained 56
  in graphs 96
  in reports 106
  labelling 49
  protecting 50
  selecting 22
ranges in @ functions 34
RECALC 375
Recalc All command 160
Recalc Range command 160
Recalculate 13
Recalculate command 159
recalculation 85
recalculation, automatic and manual 45
REDCOLOR 301, 391
Redo command 146
references, circular 45
REGRESSION 375
REPEAT 247
REPLACE 247
REPORT 376
Report Inspector 68, 69
Report Layout command 155
Report Layout Inspector 102
Reports 68
reports
  adding borders 109
  changing page size 109
  colors in tags 112
  column borders 110
  creating a new report 106
  customizing tags 111
  defining ranges 106
  fonts in tags 112
  formatting 41
  headers and footers 110
  naming 105
  page breaks 41
  printing 114
  row borders 109
  setting margins 107
  special tags 41, 110
  tag formatting code table 42, 112
reports, printing 373, 376
REPT 247
Reset Column Size command 154

Reset Row Size command 154
RESETCOLSIZE 343
RESETROWSIZE 342
resizing cells 87
resizing rows and column 55
resizing the worksheet 84
Revert to Saved command 140
Ribbon Chart Command 178
RIGHT 248
Right command 152
ROOT 217
ROUND 217
Row and Column Size menu 154
row height, changing 37, 87
row size, changing 342, 358, 361
row size, getting 389
ROWS 270
rows, hidden 339
rows, hiding 340, 362
Run Macro command 161

S
SAME 271
SAME() 58
SAVE 319
Save As command 140
Save command 139
saving your work 38
scale 93
Scatter Chart command 171
scatter charts 98
SCENARIO 376
Scenario Inspector 124, 129
scenarios
  double input 127
  single input 126
Scientific command 153
SCIENTIFICFORMAT 302
scripts 37
scripts, writing 161
SECOND 282
SELECT 331
Select All command 146
SELECTALL 331
selecting
  a cell 33
  a range 22, 34
  all cells 54
  large ranges 54
  rows and columns 54
Send to Back command 182
SENDTOBACK 367
Set Current Window Tools as Default command 183
SETADDRESS 377
SETALIGN 344
SETALTCOLOR 272, 345
SETALTPATTERN 346
SETBKGCOLOR 273, 347
SETBKGPATTERN 348
SETBORDER 348
SETBORDERCOLOR 349
SETBORDERPATTERN 350
SETBORDERTYPE 351
SETCOLOR 274, 352
SETCOLSIZE 353
SETDISPLAYFMT 354
SETFMT 354
SETFONT 355
SETGRAPHSIZE 367
SETGRAPHXY 368
SETINPUT 377
SETPATTERN 356
SETPRECISION 357
SETROWSIZE 358
SETTEMPLATE 359
setting type 351
SETVERTALIGN 360

SGN 218
Shape Panel 101
Shapes 157
Sheet Inspector 83
Sheet menu 159
sheet size 73
Show Menus 138
SIGN 218
SIGNAL 275
SIN 218
Single Border command 153
Single Input Scenarios 126
sizing a worksheet 25
sizing rows and columns 55
Slash Help command 138
SLN 237
Smaller Font command 151
Smart Column Size command 154
Smart command 152
Smart Count command 163
Smart Fill 10
Smart Max command 162
Smart menu 162
Smart Min command 162
Smart Row Size command 154
Smart Std Dev command 163
Smart Variance command 163
SMARTCOLSIZE 361
SmartFill 10, 11, 12, 64
SMARTROWSIZE 361
Smartsum command 162
SOLVE 378
SORT 379
Sort command 164
sorting data 48
Special Constants 304
Special Features 7
Special Functions (non-@) 251
special keys 2
spreadsheet
  column width 36
  cursor controls 31
  default size 25
  editing data 31
  entering data 31
  filename 30
  moving around 31
  ranges 33
  row height 37
  saving 38
  sizing 25
SQL 4
SQL query, running 374
SQRT 219
Stacked Bar Chart command 169
Stacked Column Chart command 170
std dev, smart 163
STDDEV 220
STDEV 220
Stone Design 4
Stored Procedures 118
string concatenation 193
string data 31
String Functions (non-@) 240
String Input command 147
string input, clearing 323
Strings 189
STRTONUM 332
style template
  getting 390
Style Template menu 155
style templates 76
  setting 359
  setting to defaults 337
Subroutines 314
Subtract command 145
SUM 222
sum, smart 162

Summary 70
Sybase query, running 374
SYD 238
SYSTEM 380

T
table of operators 312
TAN 222
Technical Support 5
technical support 17
template
  getting 390
templates
  setting 359
  setting to defaults 337
THISCOLUMN 305
THISROW 305
three-dimensional graphs 100, 177, 178, 179, 180
TIME 283
TIMEVALUE 283
TINTL1FORMAT 303
TINTL2FORMAT 303
titles on graphs 92
To Lower Case command 148
To Proper Case command 148
To Upper Case command 148
TODAY 303
Tool Bar 7
Tool Holder 27
Toolbar command 158
tools 27
Tools on the Worksheet 9
TRANSPOSE 332
Transpose command 145
TRIM 248
TRUE 305
Tutorial 60
tutorial
  advanced topics 42
  basic topics 17
  print topics 38
typographical conventions 2

U
UNDERALL 368
Undo command 146
Unhide Columns command 155
Unhide Rows command 155
UNHIDECOL 362
UNHIDEROW 362
units of measure 73
UPPER 249, 333
USERNAME 305

V
VALUE 249
values, copying 326
VAR 223
variable assignment 310, 372
Variables 309
variance, smart 163
VBREAK 363
VLOOKUP 276

W
WEEKDAY 284
Welcome 1
what-if tables 124
WHITECOLOR 301, 391
width of columns, changing 36
windows in worksheets 44
Windows menu 182
WORKDAY 284
worksheet
  base styles 58
  column width 36
  cursor controls 31
  default size 25

display styles 58
editing data 31
entering data 31
filename 30
moving around 31
name, changing 318
name,changing 319
new 139
ranges 33
resizing 25, 84
row height 37
saving 38, 319
worksheet titles, non-scrolling 44
worksheet windows 44
Wrap command 152

X
XOR 312
XY Chart command 172
XY charts 99

Y
YEAR 285
YELLOWCOLOR 301, 391