User's Guide

Mobile Excel® 2004

for Palm® OS

II *Mobile*Systems®

http://www.mobi-systems.com/

Contents

Getting Started

System Requirements

- **Version 3.0 of HotSync** or higher, available as a free upgrade from Palm, Inc.: www.palm.com (and included in the Desktop 3.0 upgrade, also available at the same website, and Desktop 2.6.1-2.6.3 available at http://support.handspring.com.

- **Windows 98, Windows ME, Windows 2000, Windows NT** and **Windows XP**

- Mobile Excel will import workbooks created only with Microsoft Office 97 or higher versions of the office suite.

Installation and Uninstallation

Installation on Windows PCs

1. Download MobileExcel.exe file into a selected folder on your PC.
2. Run the exe file, which is a self-extractable installation tool for the Mobile Excel Palm application and Mobile Excel HotSync conduit.
3. Follow the installation instructions.
4. Put your Palm device in its cradle, connect it to the PC and synchronize to install the program on the handheld device.

Uninstalling Mobile Excel from your Palm device

1. Tap **Menu>Applications >Delete** from the *Main Application Launcher* on your PDA.
2. Select Mobile Excel from the list and then **Delete.**
3. Confirm the deletion when prompted to remove the program from your Palm device.

Uninstalling Mobile Excel from your Desktop

1. Click **Start> Settings> Control Panel.**

2. Double-click the *Add /Remove Programs* icon.

3. Select Mobile Excel 2004 from the list of installed programs and click **Remove.**

4. Follow the instructions on your screen to completely remove Mobile Excel from your desktop computer.

Installing Using a Macintosh

The distribution files for installation through Mac OS computers come as **sit archive files**. The sit files are not Palm readable files but archive files that contain the installation files. To be able to extract these files you will need **StuffIt Expander**. The expander program is free and can be downloaded from http://www.stuffit.com/win/expander/index.html.

To extract the sit archive and install the downloaded software on your Palm device:

1. Expand the downloaded .sit file and save the extracted files in selected folder on your desktop computer.

2. Double-click the HotSync Manager icon in the Palm folder.

3. Choose Install Handheld Files from the HotSync menu.

4. From the User pop-up menu, select your user name.

5. Click Add to List.

6. Check if the folder with the extracted files does not have subfolders with additional databases and specific for your device files and instructions. If necessary, add those files to the install tool. The main folder can also contain text help file that can help you in deciding which files should be installed on your Palm device.

7. Select the folder where you have extracted the distribution sit file from the popup menu.

The files in Palm readable format have **pdb** or **prc** file extensions. The files that can be installed to your handheld device have these file extensions. Any other files will not be recognized by the HotSync tool and will generate an error. Most generally the prc files are actual Palm OS programs while the pdb files contain data.

---

8. Select all files with prc and pdb extensions.

9. Click Add File to add the selected files to the Install Handheld Files List.

10. Check if the folder with the extracted files does not have subfolders with additional databases and specific for your device files and instructions. If

necessary, add those files to the install tool. The main folder can also contain text help file that can help you in deciding which files should be installed on your Palm device.

11. Close the Install Handheld Files window.

12. Put your Palm device in its cradle, connect it to the PC and synchronize.

Registration

Mobile Excel is a downloadable shareware product available for free evaluation period or registration at our home site: www.mobi-systems.com

The trial period is 15 days, after which you should register the product to be able to use it.

To register your copy of Mobile Excel, order the product at http://www.mobi-systems.com and use the online registration form, link to which you will find only in your purchase confirmation message, to obtain the **Unlock key** for your product

Product IDs and Unlock Keys

The **Product ID** is 4x6 symbols sequence, used for the registration key generation. To obtain the **Product ID** for your copy of Mobile Excel, start the program and tap **Enter Code**. The *Registration Form* will appear, in which you will find the required ID.

The **Unlock Key** is a combination of 4x6 symbols (for example: **3C3125-3AB9D8-DE5203-F4345B**), which you will receive after you submit your **Product ID** either through the online registration form or by e-mail to support@mobi-systems.com.

Use the product registration form above to input the **Unlock Key** and complete your registration. To enter your **Unlock Key**:

1. Go to Menu> **Help**> **Registration**.

2. Type in your **Unlock Key** in the space provided in Mobile Excel registration screen.

3. Tap on **OK** to continue or **Cancel** to abort the entry.

Online Registration of Products

After the successful completion of an online purchase, you will receive a confirmation message for your order. In this message you will find link to online registration resources for your purchased products. Input the **Product ID** for the particular product in the corresponding field and click the [*ENTER PRODUCT ID*] button. Your **Unlock Key** will be e-mailed to you shortly.

You can access the **Unlock Keys** for your registered products at any time through the online registration form and also obtain second key in cases of PDA change.

Keep your purchase confirmation message since the link to the online registration resource is available only there. In case you lose the message, visit http://www.mobi-systems.com/lost-registration.asp and fill out the form provided to re-request the order confirmation message. The message will be resend to your e-mail address shortly.

Working with Workbooks

A workbook is a file that contains one or more worksheets, which you can use to organize various kinds of related information. You can enter, edit data and perform calculations based on data from more than one worksheet.

To help you easily manage your workbooks, each time you start Mobile Excel, first the *workbook launcher* will appear. The *launcher* will show all Mobile Excel files currently available in the main memory of your Palm device and also all Mobile Excel files installed on inserted memory cards.
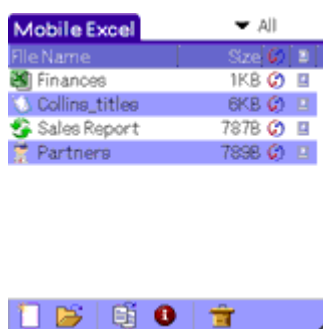
Immediately after the workbook name, the *launcher* will show the size of the file, its synchronization status and location.

To sort the workbook list by any column, just tap the title bar for the selected column. For example, to sort the files by File Name, just tap the title bar for that column. The files will be ordered in ascending list. A second tap on the title bar will reverse the sorting to descending.

Scheduling Workbooks for Synchronization

For files that are scheduled to be synchronized, the *launcher* will show icon, while workbooks that will not be synchronized will appear with icon. To change the synchronization status of a workbook, just tap on its synchronization icon to switch on and off the synchronization. Accordingly, the icon will change indicating the current status of the file.

Please note that workbooks on memory cards cannot be synchronized.

File Locations- Working with Memory Cards

Files which are in the main memory of your Palm device will appear with ▢ icon the last column in the *launcher.* Any other Mobile Excel file placed on memory expansion card will be with ▢ icon. To change the location of a workbook and move its file from the main memory to a memory card and vise versa, just tap the *location icon* for the workbook. The *location icon* will change accordingly.

Please note that when you move a workbook on memory card, its file is placed in folder **\Palm\Programs\MobileExcel.** To enhance the application speed, Mobile Excel will also list in the launcher only workbooks saved in this default directory.

To load workbooks saved in folder different from the default one or manage files on memory cards, use the **Load from Card** option in the *File menu group*.

To open a folder, just tap on its name. A second tap on opened folder will close it.

Use the horizontal and vertical scroll bars to move left and right, and up and down in the directories structure.

When you locate the desired file or whole directory, select it and tap the **Load** button.

To delete an empty folder or file from your memory card, select the file or directory from the list and then the **Delete** button.

The **Create** button allows you to add new folders. To create a new directory, select the parent folder where you want to place the new one and type in the full folder name with included path. Tap the **Create** button.

To close the *Load File* form and return to the *workbook launcher*, tap the **Cancel** button.

NATIVE MICROSOFT EXCEL XML FILES

Mobile Excel allows you to open and modify native Mobile Excel files saved in XML format that are installed directly on the memory card of your PDA.

To install a Microsoft Excel file on a memory card:

1. Open the file on your desktop computer and select **Save as** from the *File menu* in Microsoft Excel.

2. Select **XML Spreadsheet (*.xml)** as *Save as type.*

3. Open the Palm desktop Install Tool and add the new created XM file to the list of files to be installed on your Palm device.

4. In the list of files to be installed in the Palm synchronization software, select the XML file and click **Change Location**.

5. Move the file to the list of files that should be installed on your memory card.

6. Synchronize with your PDA. All XML files will be installed directly to the default Mobile Excel directory and thus be accessible in the *workbook launcher* the next time you start Mobile Excel.

Managing Workbooks

Mobile Excel toolbar which is available while you work in the *workbook launcher* provides quick single-tap access to the most frequently used functions performed over whole workbooks. To hide and show back the toolbar, just tap the right down corner of the working area, indicates with a small triangle.

NEW

To create a new workbook, just tap the icon. The *New File* from will appear. Type in the name for the new workbook in the space provided. Select a category for the file and if you wish, tap on its default icon to open the library of predefined icons. Double-tap an icon in order to assign it to the created workbook.
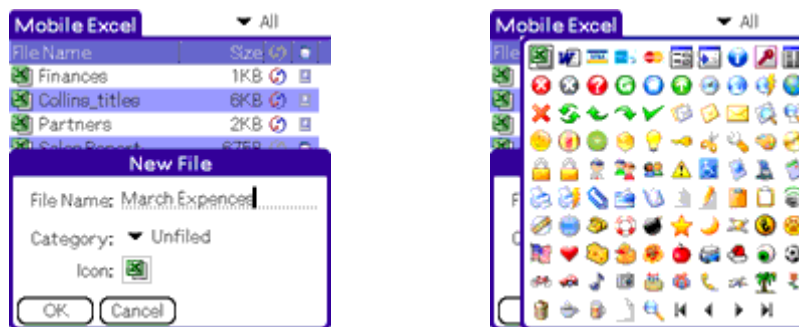
Tap **OK** to close the *New File* dialogue and create the workbook. After the file is created it will be automatically opened so you can start to enter data in it.

OPEN

To open a workbook, just select it from the list in the *workbook launcher* and tap the icon from the toolbar. To return to the *launcher* at any time, tap the icon.

COPY

To create a copy of existing workbook, select the source file from the list and tap the
icon. The new workbook will be named **Copy of <source file name>.** Its name
can be changed through the *workbook info* dialogue, accessible through the icon
and described in the next section. The copied workbook will contain all worksheets
and data that the source one has.

WORKBOOK INFO

The *workbook info* form allows you to change the name, category or icon of existing
files, as well as provides information about the file size and the date of creation of the
workbook. To open the *workbook info,* just tap the icon.

When you have finished with the workbook properties changes, just tap the **OK** button
to close the *workbook info* form and save the modifications, or alternatively- **Cancel** to
close the dialogue window without saving the changes.

DELETE

To delete a workbook, just select it from the list in the *workbook launcher* and tap the
button from the toolbar. Mobile Excel will ask you to confirm the deletion before
actually removing the file.

Please note that is a copy of the deleted file exists in the Mobile Excel
synchronization folder on your desktop computer, the workbook will be restored
after the first synchronization. To permanently delete a file, you have to remove it
from both Mobile Excel and the desktop sync folder.

Categorizing Workbooks

To facilitate your work with large number of workbooks, Mobile Excel allows you to
organize workbooks by putting them in categories.

To filter just the workbook from a selected category in the *workbook launcher* list:

1. Tap the menu trigger in the up right corner of the launcher working area. A popup menu with all categories will appear.

2. Select category from the menu.

By default Mobile Excel shows the workbooks from all categories.

To change the category of a workbook, just select the file from the *launcher* and tap the 🛈 icon. Select the desired category from the popup list for **Category** in the *File Info* form that will appear.

Mobile Excel gives you the ability to create, rename and delete categories. The categories managing form is accessible when you tap the category trigger in the upper right corner of the *workbook launcher* form and then select **Edit Categories** from the popup list that will appear.

The buttons in the *Edit Categories* form that will appear are as follows:

⊞ - Adds new category. First type in the category name in the text field and then tap the add button.

⋯ - Renames the currently selected category with the text in the edit field.

✕ - Deletes selected category.

↑↓ - Sorts the categories list in ascending or descending order. The sort order will be kept when categories are show in the *category popup menu*.

To close the *Edit Categories* dialogue, just tap the **Done** button.

Setting Workbook Preferences

Through the *Workbook Preferences* form you can change the appearance of the *workbook launcher.* The preferences also allow you to specify CSV separator to be used by Mobile Excel when you synchronize with Microsoft Excel versions that do not support XML files (e.g. Microsoft Excel 97 and 2000).
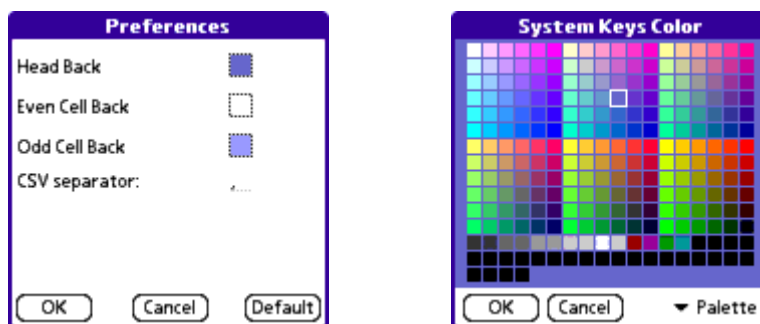
o **Header Background** – through this option you can change the background color of the column title bar. To change the background color, just tap the color box. The *System Keys Color* form will appear. Select a new color and tap OK.

o **Even Cell Background** – the background used for cells from even rows and columns. You can change the default background color as described in the Header Background bullet.

o **Odd Cell Background** – the background color for cells from odd rows and columns.

o **CSV separator** – this field is used when you want to synchronize with versions of Microsoft Excel that do not support XML files. In that case, make sure that the **CSV Separator** is the same that you use on your desktop computer. In most Windows versions you can check your **list separator** when you open the Regional Options in the Control Panel and switch to the *Number tab.*

Working with Spreadsheets

The primary document that you use in Mobile Excel to store and work with data is called *worksheet*. A worksheet consists of cells that are organized into columns and rows. A worksheet is always stored in a workbook.

Mobile Excel supports unlimited number of worksheets in a single workbook. The name of the opened worksheet is shown in the upper left corner of the working area.

You can switch between the different worksheets in the currently opened workbook through the *worksheet popup menu* that appears when you tap the trigger in the upper right corner of the screen.

Managing Spreadsheet

In Mobile Excel spreadsheets can be created, renamed and deleted through the *Edit Worksheet* form. To open the spreadsheet manager, just tap the *spreadsheet menu trigger* in the up right corner of the working area and select **Edit Sheets** from the popup list that will appear.
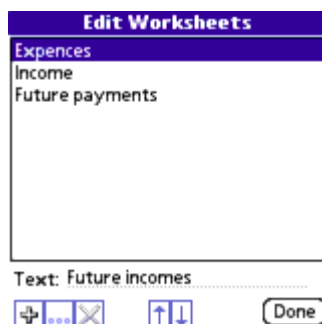
RENAME

To rename a worksheet:

1. Open the *Edit Worksheets* manager as described above.

2. Select the worksheet from the list.

3. Type in the new name in the text edit box of the *Edit Worksheet* form.

4. Tap the ⬚⬚⬚ icon. The worksheet's name will be changed and the new name will appear in the *Edit Worksheet* manager.

INSERT

To insert a new worksheet in the currently opened workbook, just open the Mobile Excel main menu and select **Worksheet** from the *Insert menu group*. The following dialogue form will appear:

1. Specify the position where the new workbook should be placed by selecting a check box:

   o Before current

   o After Current

   o At the end

2. Type in a name for the worksheet in the corresponding text field.

3. Click **OK** to create the worksheet or Cancel to dismiss the creation. The new worksheet will be created and Mobile Excel will automatically switch to it so you can start entering data.

Alternatively, you can create new worksheets through the *Edit Worksheets* form. To add a new spreadsheet:

1. Open the *Edit Worksheets* manager.

2. Type in the name for the new worksheet in the text edit field.

3. Tap the ⊞ button.

To delete a worksheet:

1. Open the *Edit Worksheets* manager as described above.

2. Select the worksheet from the list.

3. Tap the ⊠ icon.

SAVING WORKSHEETS

Whenever you try to exit a workbook, either by clicking the 🏠 icon or the application button of the graffiti area of your Palm device, Mobile Excel will ask you whether to save the changes made.

If you want to save a workbook while you are working, you can use the **Save** option in the *File menu group.*

To save a workbook under a new name, open the *File menu* and select **Save As.**

o Enter a new name for the workbook in the text edit field for **File Name.**

o Select the type under which you want to save the workbook – as a XML file or as CSV.

If you synchronize with Microsoft Excel XP or Excel 2003, save the file as a XML file. To be able to synchronize with Microsoft Excel 97 or 2000, save the file as CSV.

---

o Select location where you want to save the file – on the Handheld or on memory card. If you select to place the file on a memory card, Mobile Excel will allow you also to specify the path on your memory card where you want so save the worksheet. By default the program will suggest the default Mobile

Excel folder **\Palm\Programs\MobileExcel** on the memory card. You can change the default path through the **Path** edit field.

To close a workbook without saving the changes, open the *File menu* and select **Close.**

Zooming Worksheets

On Palm devices with Palm OS 5 or higher version, Mobile Excel supports true type fonts. Sample true type fonts come with the distribution zip or sit archive for Mobile Excel, or will be installed on your Palm during the installation setup program for Windows.

When Mobile Excel is able to locate true type fonts on your handheld, an additional zoom option will be enabled in the program. The zoom option can be accessed through the zoom percent icon that appears in the upper part of the worksheet screen next to the home icon.
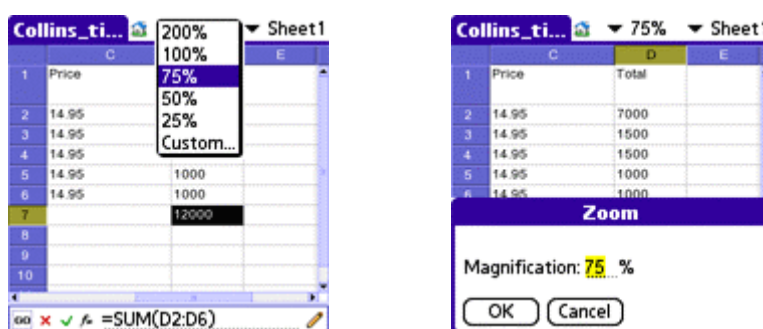
The supported zoom sizes are 25%, 50%, 75% (the default mobile Excel size), 100%, and 200%. In addition, through the **Custom** option in the zoom popup menu, the program allows you to specify custom zoom percent ranging from 25% up to 300%.

Worksheet Preferences

The worksheet preferences allow you to customize the way worksheets are shown- set background colors for the headers and cells, move the edit cell bar, show and hide the gridlines.
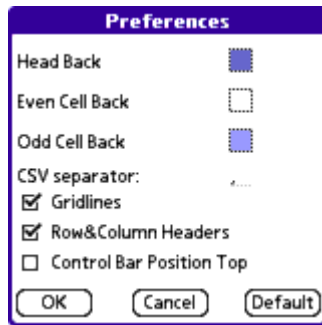
The color settings are similar to that described in Workbook Preferences chapter.

- o **Gridlines** – through this check box you can specify whether Mobile Excel will show gridlines when visualizes worksheets or not.

- o **Row&Column Headers** – through this checkbox you can hide and show the headers of the columns and rows in spreadsheet view.

- o **Control Bar Position Top** – this option allows you to change the position of the *cell edit control bar.* To position the bar on the top above the columns header, enable the check box.

To restore the default settings, just tap the **Default** button.

To save the changes that you have made in the Preferences form, tap the **OK** button. Alternatively, you can discard the changes by tapping the **Cancel** button.

Working with Columns and Rows

For easier data management Mobile Excel allows certain operations to be performed over whole columns and rows.

To select a row or column, just tap on its index name in the horizontal for columns and the vertical for rows title bar. The whole column or row will be marked in black indicating that it is currently selected.

While a column or row is selected, Mobile Excel provides a popup menu for quick access to the most frequently used functions over columns and rows. To open the popup functions menu, select the desired object and fold the stylus on its title bar for a couple of seconds.
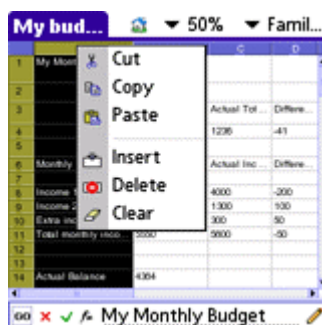
Resize

To resize a column or row, place the stylus on its border and drag in the desired direction until the object reaches the size that you wish.

Insert

Mobile Excel allows you to insert single rows and columns or to add multiple row or columns at a time.

To rows or columns:

1. Select the row or column before which you want to place the new item.

2. Hold the stylus on the selected column or row title bar for a couple of seconds until a popup menu appears.

3. Select **Insert** from the popup menu list. The following dialogue form will appear:

4. Type in the number of columns or rows that you want to insert.

5. Tap **OK** to add the columns or rows.

An alternative way to insert columns or rows is selecting the **Columns** or **Rows** options from the Mobile Excel *Insert menu group.* The row or column before which you want to add the new items has to be selected beforehand.

Delete

The deletion of columns and rows is similar to the insertion of new columns. Select the row or column that you want to delete, hold the stylus on the objects title bar until popup menu appears, and select **Delete** from the popup list.

The function can be applied also to a selection of rows or columns.

Cut, Copy, Paste, and Clear

All these functions can be applied to selection of rows or columns. The functions are accessible either through the *Edit Menu group* or through the *function popup* menu that appears when you hold the stylus on the title bar of the selected objects.

o Cut – saves the contents of the selection in the systems clipboard and the cut selection area is deleted. The operation is used altogether with Copy and Paste.

o Copy – the contents of the system clipboard is copied in the selected area. The copy selection area must be of the same type- rows or columns- as the data in the clipboard. The operation is usually used after Cut as well as in combination with Paste.

o Paste- Copies the contents of the system clipboard usually selected with Copy or Cut, in the highlighted area of rows or columns. The size of the paste selection area should be the same as the contents of the clipboard.

o Clear- Deletes the contents of the selected area.

These functions can also be applied to selection of cells.

Working with Cells

Select Cell or Multiple Cells

To select **a single cell**, tap in the cell box. After a cell is selected, its contents will also appear in the *edit field* of the *cell toolbar*.

To select **text in a cell,** first select the cell, tap in *edit box* in the *cell toolbar*, and then select the text in the cell.

To select **a range of cells**, tap the first cell of the range, and then drag to the last cell.

To select **all cells in a worksheet,** tap the **Select All** button or alternatively choose **Select All** from the *Edit menu group*. The **Select All** button is in the upper left corner of the worksheet area and in the intersection of the rows and columns title bars.

The Cell Toolbar

The functions from the cell toolbar are described below in the order of their appearance:

o   **GO** - Shortcut to the GoTo function, which is described in the next chapter.

o   ✗ - Restores the original contents of the cell.

o   ✓ - Confirms the changes made to the cell.

o   *ƒ✻* - Opens the *Select Function* form that allows you to embed function within a cell.

o   edit field – allows you to edit the contents of the selected cell.

o   ✐ - opens an extended edit form with built-in virtual keyboard.

Go To

The Go To function allows you to directly move to a cell by specifying its row and column. To open the *Go To dialogue form*, open the *File menu group* and select the **Go To** option.
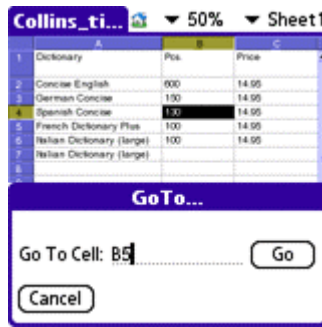
The only valid reference to a cell starts with the cell's column, directly followed by the cell's row.

The function can also be started through the **GO** icon in the *cell toolbar*.

Find and Replace

The find function allows you to search for text within cells or whole cells within a spreadsheet and optionally to replace the found result with a new string.

To open the *Find / Replace dialogue,* open the *File menu* and select **Find** from the menu options**.**
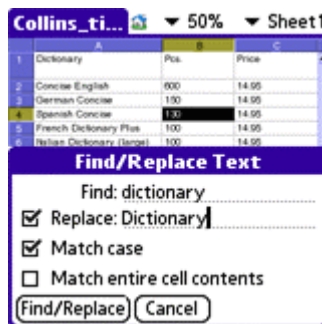
Each of the fields and options in the *Find/Replace Text* form are described below:

- o **Find** – type in the find string here. By default Mobile Excel will offer you the contents of the last selected cell in the worksheet.

- o **Replace** – enable the check box for Replace only if you want to use this option. Otherwise the function will act as a simple find and no data will be replaced. You will also have to specify a **replace string** to be used.

- o **Match case** – enable this check box if you want an exact case sensitive match after the find function.

- o **Match entire cell contents** – enable this check box if you want to disable the searching within cell text and would like to receive only whole cells matching the find criteria.

Set the find preference and tap the **Find/Replace** button. If the search returns any matches, the first cell satisfying the find criteria will be selected (or replaced). To continue the search and move next cell that matches the search condition, select **Find Next** from the *Edit menu.*

Cut, Copy, Paste, and Clear

These operations can be applied to a single cell, a range of cells, and even a whole spreadsheet. The functions are described in details in chapter Working with Spreadsheets. It is important for the copy and paste operations that the copied data should be with the same size as the selected area where you want to place the contents of the clipboard.

Working with Functions

Functions are predefined formulas that perform calculations by using specific values, called arguments given in a fied order. Functions can be used to perform simple or complex calculations.

Functions can be embedded in worksheet cells. Each function begins with an equal sign (=), followed by the *function name*, an opening parenthesis, the *arguments* for the function separated by commas, and a closing parenthesis.

The arguments can be numbers, text, logical values such as TRUE or FALSE, arrays, or cell ranges. Arguments can also be constants, formulas, or other functions.

Nested Functions

In certain cases, you may need to use a function as one of the arguments of another function. For example, the following formula uses a nested SUM function and compares the result with the value 30.

=IF(SUM(A1:A7)>30,SUM(B1:B7),C1)

When a nested function is used as an argument, it must return the same type of value that the argument uses.

Inserting a function

To insert a function:

1. Select the cell where you want to place the function.

2. Select the function icon f_* from the *cell edit toolbar*. The *Select Function form* will appear.
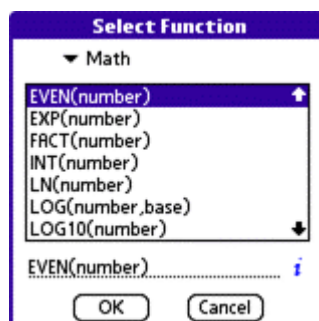
3. Use the Function type popup menu to filter just the functions that you need and easily locate the desired one.

4. Scroll in the functions list until you locate the function that you need.

5. You can view help for each function by tapping the $\boldsymbol{i}$ icon.

6. When you find the desired function, select it and tap the OK button.

7. The function will be inserted in the cell edit box and now you can provide its arguments. You can either enter the arguments manually or use the stylus to select cells from the worksheet as arguments.

   Each next selected cell will be added as a separate argument separated with comma from the previous one.

   To give cells range as an argument of the function, just tap on the first cell from the range and drag in the desired direction until the whole range is selected.

If you are sure about the syntax of a function and its arguments, you can skip steps 2-6 and type in the function directly in the *cell edit box* in the *cell toolbar*. For instance, you can directly type in =SUM(A1:A6).

---

8. When you have finished entering the function, tap the ✓ icon to execute it.

Formula Errors

Below is a table of the error messages that Mobile Excel can display in case there is something wrong with a functions arguments.

FORMULA ERROR	DESCRIPTION
Error evaluating formula. Out of memory.	Insufficient memory for calculating the formula. Please check the available free space on your Palm device and try to free some space.
Error. Invalid formula.	Incorrect formula syntax. Check if the formula starts with =, +, or -.
Formula error. Missing operand.	Missing operator. For instance =5%10 will result in this error because there is not operator mathematical operator. The correct formula is '=5+%10' or '=5-%10'.
Formula error. Missing operator.	Missing argument. For example, '=5 2'. Correcy formulas are '=5+ 2'. '=SUM(5,2)', etc.
Formula error. Illegal token.	Example: '= 5+4)' – missing left parenthesis Check the formula syntax and correct the error.
Formula error. Missing quote (").	Missing quotations. Example: '="ala" + "bala ' Check the formula syntax and correct the error.
Formula error. Missing array closing bracket. "] ".	Example: (**}**): '={5, 3' The array closing bracket is missing. Check the formula syntax.

	formula syntax.
Formula error. Illegal identifier.	Illegal formula argument. Example: '=SUM(ABCDEF)' The provided agruement is neither reference to existing cell, nor predefined name.
Formula error. Inconsistent array.	Arrays used as function arguments should have the following syntax: '{a1,a2,a3;b1,b2,b3;…}' The items in each array row are separated by comma and the different rows are separated by semi-column. This error message indicates incorrect array definition.
Formula error. Missing array element.	Incorrect array definition. This error indicates that array item has been omitted. Example: '={1,2, ,4}' The third array item is omitted.
Formula error. Illegal array element.	Incorrect array definition. This error indicates that invalid array item has been provided. Arrays can consist of numbers, errors (e.g. #REF!), and logical values (true/false). References cannot be used as array items. Example: '= {1,2,A1}' The last item in the array is a reference to cell A1.
Formula error. Missing function closing bracket ") ".	Check the function syntax. A closing parenthesis is missing. Example: '=SUM(A1:B2'
Formula error. Too deep function nesting.	The function nesting is on more than 7 levels.
Formula error. Unknown function.	Unknown function. Check the function syntax and name. Example: '=ALABALA(5)'
Formula error. Missing single quote (').	This error message indicated that an opening or closing apostrophe is missing. Example: '= 'Sheet 1!A1' (the apostrophe between 1 and ! is omitted)
Formula error. Circular reference.	This error indicates a circular expression that cannot be calculated. Example: A1 '=B1' and B1 '=A1'
Unknown evaluation error.	Evaluation error. Check the syntax of the formula and recalculate the formula.
Formula error. Not enough arguments.	Invalid formula syntax. Not enough argements were provided for the given formula. Check the formula syntax and provide the necessary arguments.
Unknown formula error.	
Formula error. Too much arguments.	Invalid formula syntax. Too much function arguments are provided. Check the formula syntax and provide the necessary arguments.

Function Descriptions

Math and Trigonometry

ABS	Returns the absolute value of a number. The absolute value of a number is the number without its sign.	*ABS(number)* Number is the real number of which you want the absolute value.	=ABS(2) Absolute value of 2 (2)
ACOS	Returns the arccosine, or inverse cosine, of a number. The arccosine is the angle whose cosine is *number*. The returned angle is given in radians in the range 0 (zero) to pi.	*ACOS(number)* Number is the cosine of the angle you want and must be from -1 to 1.	=ACOS(-0.5) Arccosine of -0.5 in radians, 2*pi/3 (2.094395)
ACOSH	Returns the inverse hyperbolic cosine of a number. Number must be greater than or equal to 1. The inverse hyperbolic cosine is the value whose hyperbolic cosine is *number*, so ACOSH(COSH(number)) equals *number*.	ACOSH(number) Number is any real number equal to or greater than 1.	=ACOSH(1) Inverse hyperbolic cosine of 1 (0)
ASIN	Returns the arcsine, or inverse sine, of a number. The arcsine is the angle whose sine is number. The returned angle is given in radians in the range -pi/2 to pi/2.	ASIN(number) Number is the sine of the angle you want and must be from -1 to 1.	=ASIN(-0.5) Arcsine of -0.5 in radians, -pi/6 (-0.5236)
ASINH	Returns the inverse hyperbolic sine of a number. The inverse hyperbolic sine is the value whose hyperbolic sine is *number*, so ASINH(SINH(number)) equals *number*.	*ASINH(number)* Number is any real number.	=ASINH(-2.5) Inverse hyperbolic sine of -2.5 (-1.64723)

ATAN	Returns the arctangent, or inverse tangent, of a number. The arctangent is the angle whose tangent is *number*. The returned angle is given in radians in the range -pi/2 to pi/2.	*ATAN (number)* Number is the tangent of the angle you want.	=ATAN(1) Arctangent of 1 in radians, pi/4 (0.785398)
ATAN2	Returns the arctangent, or inverse tangent, of the specified x- and y-coordinates. The arctangent is the angle from the x-axis to a line containing the origin (0, 0) and a point with coordinates (x_num, y_num). The angle is given in radians between -pi and pi, excluding -pi.	*ATAN2(X,Y)* X is the x-coordinate of the point and Y is the y-coordinate of the point.	=ATAN2(1, 1) Arctangent of the point 1,1 in radians, pi/4 (0.785398)
ATANH	Returns the inverse hyperbolic tangent of a number. Number must be between -1 and 1 (excluding -1 and 1). The inverse hyperbolic tangent is the value whose hyperbolic tangent is *number*, so ATANH(TANH(number)) equals *number*.	*ATANH(number)* Number is any real number between 1 and -1.	=ATANH(0.76159416) Inverse hyperbolic tangent of 0.76159416 (1, approximately)
COS	Returns the cosine of the given angle.	*COS(number)* Number is the angle in radians for which you want the cosine.	=COS(60*PI()/180) Cosine of 60 degrees (0.5)
COSH	Returns the hyperbolic cosine of a number.	*COSH(number)* Number is any real number for which you want to find the hyperbolic cosine.	=COSH(4) Hyperbolic cosine of 4 (27.30823)
COUNTIF	Counts the number of cells within a range that meet the given criteria.	*COUNTIF(range,criteria)* Range is the range of cells from which you want to count cells. Criteria is the criteria in the form of a number, expression, or text that defines which cells will be counted.	=COUNTIF(B2:B5,">55") Number of cells with a value greater than 55 in the second column above (2)
DEGREES	Converts radians into degrees.	*DEGREES(angle)* Angle is the angle in radians that you want to convert.	=DEGREES(PI()) Degrees of pi radians (180)

EVEN	Returns number rounded up to the nearest even integer. You can use this function for processing items that come in twos. For example, a packing crate accepts rows of one or two items. The crate is full when the number of items, rounded up to the nearest two, matches the crate's capacity. The function can be used only for numeric fields	*EVEN(number)* Number is the value to round.	=EVEN(1.5) Rounds 1.5 up to the nearest even integer (2) =EVEN(3) Rounds 3 up to the nearest even integer (4)
EXP	Returns **e constant** raised to the power of number. The constant e equals 2.71828182845904, the base of the natural logarithm. EXP is the inverse of LN, the natural logarithm of number.	*EXP(number)* Number is the exponent applied to the base e.	
FACT	Returns the factorial of a number. The factorial of a number is equal to 1*2*3*...* number.	*FACT(number)* Number is the nonnegative number you want the factorial of. If number is not an integer, it is truncated	=FACT(5) Factorial of 5, or 1*2*3*4*5 (120)
INT	Rounds a number down to the nearest integer.	*INT(number)* Number is the real number you want to round down to an integer.	=INT(8.9) Rounds 8.9 down (8) =INT(-8.9) Rounds -8.9 down (-9)
LN	Returns the natural logarithm of a number. Natural logarithms are based on the constant e (2.71828182845904). LN is the inverse of the EXP function.	*LN(number)* Number is the positive real number for which you want the natural logarithm.	=LN(86) Natural logarithm of 86 (4.454347) =LN(2.7182818) Natural logarithm of the value of the constant e (1)
LOG	Returns the logarithm of a number to the base you specify.	*LOG(number,base)* Number is the positive real number for which you want the logarithm. Base is the base of the logarithm. If base is omitted, it is assumed to be 10.	=LOG(10) Logarithm of 10 (1) =LOG(8, 2) Logarithm of 8 with base 2 (3)

LOG10	Returns the base-10 logarithm of a number.	*LOG10(number)* Number is the positive real number for which you want the base-10 logarithm.	=LOG10(10) Base-10 logarithm of 10 (1) =LOG10(1E5) Base-10 logarithm of 1E5 (5)
MOD	Returns the remainder after number is divided by divisor. The result has the same sign as divisor. The MOD function can be expressed in terms of the INT function: `MOD(n, d) = n - d*INT(n/d)`	*MOD(number,divisor)* Number is the number for which you want to find the remainder. Divisor is the number by which you want to divide number.	=MOD(3, 2) Remainder of 3/2 (1) =MOD(-3, 2) Remainder of -3/2. The sign is the same as divisor (1)
ODD	Returns number rounded up to the nearest odd integer. The function can be used for numeric fields.	*ODD(number)* Number is the value to round.	=ODD(1.5) Rounds 1.5 up to the nearest odd integer (3)
PI	Returns the number 3.14159265358979, the mathematical constant pi.	*PI( )*	=PI()/2 Pi/2 (1.570796327)
POWER	Returns the result of a number raised to a power.	*POWER(number,power)* Number is the base number. It can be any real number. Power is the exponent to which the base number is raised.	=POWER(5,2) 5 squared (25)
PRODUCT	Multiplies all the numbers given as arguments and returns the product.	*PRODUCT(number1,number2,...)* Number1, number2, etc. are 1 to 30 numbers that you want to multiply.	PRODUCT(A2:A4) Multiplies the numbers contained in cells A2, A3 and A4.
RADIANS	Converts degrees to radians.	*RADIANS(angle)* Angle is an angle in degrees that you want to convert.	=RADIANS(270) 270 degrees as radians (4.712389 or $3\pi/2$ radians)
RAND	Returns an evenly distributed random number greater than or equal to 0 and less than 1. A new random number is returned every time the worksheet is calculated.	*RAND( )*	

ROUND	Rounds a number to a specified number of digits.	*ROUND(number,num_digits)* Number is the number you want to round. Num_digits specifies the number of digits to which you want to round number. If num_digits is greater than 0 (zero), then number is rounded to the specified number of decimal places. If num_digits is 0, then number is rounded to the nearest integer. If num_digits is less than 0, then number is rounded to the left of the decimal point.	=ROUND(2.15, 1) Rounds 2.15 to one decimal place (2.2)
ROUNDD OWN	Rounds a number down, toward zero. ROUNDDOWN behaves like ROUND, except that it always rounds a number down.	*ROUNDDOWN(number,num_digits)* Number is any real number that you want rounded down. Num_digits is the number of digits to which you want to round number.	=ROUNDDOWN(3.2, 0) Rounds 3.2 down to zero decimal places (3)
ROUNDU P	Rounds a number up, away from 0 (zero). ROUNDUP behaves like ROUND, except that it always rounds a number up.	*ROUNDUP(number,num_digits)* Number is any real number that you want rounded up. Num_digits is the number of digits to which you want to round number.	=ROUNDUP(3.2,0) Rounds 3.2 up to zero decimal places (4)
SIN	Returns the sine of the given angle. If your argument is in degrees, multiply it by PI()/180 or use the RADIANS function to convert it to radians.	*SIN(number)* Number is the angle in radians for which you want the sine.	=SIN(30*PI()/180) Sine of 30 degrees (0.5)
SINH	Returns the hyperbolic sine of a number.	*SINH(number)* Number is any real number.	=SINH(1) Hyperbolic sine of 1 (1.175201194)
SQRT	Returns a positive square root. If number is negative, SQRT returns an error value.	*SQRT(number)* Number is the number for which you want the square root.	=SQRT(16) Square root of 16 (4)

SUM	Adds all the numbers in a range of cells.	*SUM(number1,number2, ...)* Number1, number2, etc. are 1 to 30 arguments for which you want the total value or sum.	=SUM(A2:A4) Adds the first three numbers in the column above.
SUMIF	Adds the cells specified by a given criteria.	*SUMIF(range,criteria,sum_range)* Range is the range of cells you want evaluated. Criteria is the criteria in the form of a number, expression, or text that defines which cells will be added. For example, criteria can be expressed as 32, "32", ">32", "apples". Sum_range are the actual cells to sum.	
TAN	Returns the tangent of the given angle.	*TAN(number)* Number is the angle in radians for which you want the tangent.	=TAN(45*PI()/180) Tangent of 45 degrees (1)
TANH	Returns the hyperbolic tangent of a number.	*TAN(number)* Number is the angle in radians for which you want the tangent.	=TANH(-2) Hyperbolic tangent of -2 (-0.96403)
TRUNC	Truncates a number to an integer by removing the fractional part of the number.	*TRUNC(number,num_digits)* Number is the number you want to truncate. Num_digits is a number specifying the precision of the truncation. The default value for num_digits is 0 (zero).	=TRUNC(8.9) Integer part of 8.9 (8)

Database

The database functions have the following arguments:

o **Database** is the range of cells that makes up the list or database. A database is a list of related data in which rows of related information are records, and columns of data are fields. The first row of the list contains labels for each column.

o **Field** indicates which column is used in the function. Field can be given as text with the column label enclosed between double quotation marks, such as "Age" or "Yield," or as a number that represents the position of the column within the list: 1 for the first column, 2 for the second column, and so on.

o **Criteria** is the range of cells that contains the conditions you specify. You can use any range for the criteria argument, as long as it includes at least one column label and at least one cell below the column label for specifying a condition for the column.

DAVERAGE	Averages the values in a column of a list or database that match conditions you specify.	*DAVERAGE(database,field,criteria)*
DCOUNT	Counts the cells that contain numbers in a column of a list or database that match conditions you specify. The field argument is optional. If field is omitted, DCOUNT counts all records in the database that match the criteria.	*DCOUNT(database,field,criteria)*
DCOUNTA	Counts the nonblank cells in a column of a list or database that match conditions you specify. The field argument is optional. If field is omitted, DCOUNTA counts all records in the database that match the criteria.	*DCOUNTA(database,field,criteria)*
DGET	Extracts a single value from a column of a list or database that matches conditions you specify.	*DGET(database,field,criteria)*
DMAX	Returns the largest number in a column of a list or database that matches conditions you specify.	*DMAX(database,field,criteria)*
DMIN	Returns the smallest number in a column of a list or database that matches conditions you specify.	*DMIN(database,field,criteria)*
DPRODUCT	Multiplies the values in a column of a list or database that match conditions you specify.	*DPRODUCT(database,field,criteria)*

DSTDEV	Estimates the standard deviation of a population based on a sample by using the numbers in a column of a list or database that match conditions you specify.	*DSTDEV(database,field,criteria)*
DSTDEVP	Calculates the standard deviation of a population based on the entire population, using the numbers in a column of a list or database that match conditions you specify.	*DSTDEVP(database,field,criteria)*
DSUM	Adds the numbers in a column of a list or database that match conditions you specify.	*DSUM(database,field,criteria)*
DVAR	Estimates the variance of a population based on a sample by using the numbers in a column of a list or database that match conditions you specify.	*DVAR(database,field,criteria)*
DVARP	Calculates the variance of a population based on the entire population by using the numbers in a column of a list or database that match conditions you specify.	*DVARP(database,field,criteria)*

Financial

DDB	Returns the depreciation of an asset for a specified period using the double-declining balance method or some other method you specify.	*DDB(cost,salvage,life,period,factor)* Cost is the initial cost of the asset. Salvage is the value at the end of the depreciation. Life is the number of periods over which the asset is being depreciated (the useful life of the asset). Period is the period for which you want to calculate the depreciation. Period must use the same units as life. Factor is the rate at which the balance declines. If factor is omitted, it is assumed to be 2 (the double-declining balance

		method).
EFFECT	Returns the effective annual interest rate, given the nominal annual interest rate and the number of compounding periods per year.	*EFFECT(nominal_rate,npery)* Nominal_rate is the nominal interest rate. Npery is the number of compounding periods per year. EFFECT is calculated as follows: EFFECT= POWER((1+Nominal_rate/Npery),Npery) -1
FV	Returns the future value of an investment based on periodic, constant payments and a constant interest rate.	*FV(rate,nper,pmt,pv,type)* Rate is the interest rate per period. Nper is the total number of payment periods in an annuity. Pmt is the payment made each period; it cannot change over the life of the annuity. Typically, pmt contains principal and interest but no other fees or taxes. If pmt is omitted, you must include the pv argument. Pv is the present value, or the lump-sum amount that a series of future payments is worth right now. If pv is omitted, it is assumed to be 0 (zero), and you must include the pmt argument. Type is the number 0 or 1 and indicates when payments are due. If type is omitted, it is assumed to be 0.
IPMT	Returns the interest payment for a given period for an investment based on periodic, constant payments and a constant interest rate. For a more complete description of the arguments in IPMT and for more information about annuity functions, see PV.	*IPMT(rate,per,nper,pv,fv,type)* Rate is the interest rate per period. Per is the period for which you want to find the interest and must be in the range 1 to nper. Nper is the total number of payment periods in an annuity. Pv is the present value, or the lump-sum amount that a series of future payments is worth right now. Fv is the future value, or a cash balance you want to attain after the last payment is made. If fv is omitted, it is assumed to be 0 (the future value of a loan, for example, is 0). Type is the number 0 or 1 and indicates when payments are due. If type is omitted, it is assumed to be 0.
NOMINAL	Returns the nominal annual interest rate, given the effective rate and the number of compounding periods per year.	*NOMINAL(effect_rate,npery)* Effect_rate is the effective interest rate. Npery is the number of compounding periods per year.
NPER	Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate.	*NPER(rate, pmt, pv, fv, type)* For a more complete description of the arguments in NPER and for more information about annuity functions, see PV.

		Rate is the interest rate per period.
		Pmt is the payment made each period; it cannot change over the life of the annuity. Typically, pmt contains principal and interest but no other fees or taxes.
		Pv is the present value, or the lump-sum amount that a series of future payments is worth right now.
		Fv is the future value, or a cash balance you want to attain after the last payment is made. If fv is omitted, it is assumed to be 0 (the future value of a loan, for example, is 0).
		Type is the number 0 or 1 and indicates when payments are due.
NPV ***1** See the remarks below the table	Calculates the net present value of an investment by using a discount rate and a series of future payments (negative values) and income (positive values).	*NPV(rate,value1,value2, ...)* Rate is the rate of discount over the length of one period. Value1, value2, ... are 1 to 29 arguments representing the payments and income.
PPMT	Returns the payment on the principal for a given period for an investment based on periodic, constant payments and a constant interest rate.	*PPMT(rate,per,nper,pv,fv,type)* Rate is the interest rate per period. Per specifies the period and must be in the range 1 to nper. Nper is the total number of payment periods in an annuity. Pv is the present value — the total amount that a series of future payments is worth now. Fv is the future value, or a cash balance you want to attain after the last payment is made. If fv is omitted, it is assumed to be 0 (zero), that is, the future value of a loan is 0. Type is the number 0 or 1 and indicates when payments are due.
PMT	Calculates the payment for a loan based on constant payments and a constant interest rate.	*PMT(rate,nper,pv,fv,type)* For a more complete description of the arguments in PMT, see the PV function. Rateis the interest rate for the loan. Nper is the total number of payments for the loan. Pv is the present value, or the total amount that a series of future payments is worth now; also known as the principal. Fv is the future value, or a cash balance you want to attain after the last payment is made. If fv is omitted, it is assumed to be 0 (zero), that is, the future value of a loan is 0. Type is the number 0 (zero) or 1 and indicates when payments are due.

PV	Returns the present value of an investment. The present value is the total amount that a series of future payments is worth now. For example, when you borrow money, the loan amount is the present value to the lender.	*PV(rate,nper,pmt,fv,type)* Rate is the interest rate per period. For example, if you obtain an automobile loan at a 10 percent annual interest rate and make monthly payments, your interest rate per month is 10%/12, or 0.83%. You would enter 10%/12, or 0.83%, or 0.0083, into the formula as the rate. Nper is the total number of payment periods in an annuity. For example, if you get a four-year car loan and make monthly payments, your loan has 4*12 (or 48) periods. You would enter 48 into the formula for nper. Pmt is the payment made each period and cannot change over the life of the annuity. Typically, pmt includes principal and interest but no other fees or taxes. For example, the monthly payments on a $10,000, four-year car loan at 12 percent are $263.33. You would enter -263.33 into the formula as the pmt. If pmt is omitted, you must include the fv argument. Fv is the future value, or a cash balance you want to attain after the last payment is made. If fv is omitted, it is assumed to be 0 (the future value of a loan, for example, is 0). For example, if you want to save $50,000 to pay for a special project in 18 years, then $50,000 is the future value. You could then make a conservative guess at an interest rate and determine how much you must save each month. If fv is omitted, you must include the pmt argument. Type is the number 0 or 1 and indicates when payments are due.
SLN	Returns the straight-line depreciation of an asset for one period.	*SLN(cost,salvage,life)* Cost is the initial cost of the asset. Salvage is the value at the end of the depreciation (sometimes called the salvage value of the asset). Life is the number of periods over which the asset is depreciated (sometimes called the useful life of the asset).
SYD	Returns the sum-of-years' digits depreciation of an asset for a specified period. SYD is calculated as follows: SYD= ((cost-salvage)*(life-per+1)*2)/life*(life+1)	*SYD(cost,salvage,life,per)* Cost is the initial cost of the asset. Salvage is the value at the end of the depreciation (sometimes called the salvage value of the asset). Life is the number of periods over which the asset is depreciated (sometimes called the useful life of the asset). Per is the period and must use the same units as life.

*1 Remarks for the **NPV** function:

- Value1, value2, ... must be equally spaced in time and occur at the end of each period.

- NPV uses the order of value1, value2, ... to interpret the order of cash flows. Be sure to enter your payment and income values in the correct sequence.

- Arguments that are numbers, empty cells, logical values, or text representations of numbers are counted; arguments that are error values or text that cannot be translated into numbers are ignored.

- If an argument is an array or reference, only numbers in that array or reference are counted. Empty cells, logical values, text, or error values in the array or reference are ignored.

- The NPV investment begins one period before the date of the value1 cash flow and ends with the last cash flow in the list. The NPV calculation is based on future cash flows. If your first cash flow occurs at the beginning of the first period, the first value must be added to the NPV result, not included in the values arguments. For more information, see the examples below.

- NPV is similar to the PV function (present value). The primary difference between PV and NPV is that PV allows cash flows to begin either at the end or at the beginning of the period. Unlike the variable NPV cash flow values, PV cash flows must be constant throughout the investment. For information about annuities and financial functions, see PV.

- NPV is also related to the IRR function (internal rate of return). IRR is the rate for which NPV equals zero: NPV(IRR(...), ...) = 0.

Logical

AND	Returns TRUE if all its arguments are TRUE; returns FALSE if one or more argument is FALSE.	*AND(logical1,logical2, ...)* Logical1, logical2, ... are 1 to 30 conditions you want to test that can be either TRUE or FALSE. The arguments must evaluate to logical values such as TRUE or FALSE, or the arguments must be arrays or references that contain logical values.
FALSE	Returns the logical value FALSE.	*FALSE( )* You can also type the word FALSE directly onto the worksheet or into the formula, and the program interprets it as the logical value FALSE.
IF	Returns one value if a condition you specify evaluates to TRUE and another value if it evaluates to FALSE. Use IF to conduct	*IF(logical_test,value_if_true,value_if_false)* Logical_test is any value or expression that can be evaluated to TRUE or FALSE. This argument can use any comparison calculation operator. Value_if_true is the value that is returned if logical_test is

	conditional tests on values and formulas.	TRUE. Value_if_true can be another formula. Value_if_false is the value that is returned if logical_test is FALSE. Value_if_false can be another formula.
NOT	Reverses the value of its argument. Use NOT when you want to make sure a value is not equal to one particular value.	*NOT(logical)* Logical is a value or expression that can be evaluated to TRUE or FALSE.
OR	Returns TRUE if any argument is TRUE; returns FALSE if all arguments are FALSE. The arguments must evaluate to logical values such as TRUE or FALSE, or in arrays or references that contain logical values. If an array or reference argument contains text or empty cells, those values are ignored.	*OR(logical1,logical2,...)* Logical1,logical2,... are 1 to 30 conditions you want to test that can be either TRUE or FALSE.
TRUE	Returns the logical value TRUE.	*TRUE( )* You can enter the value TRUE directly into cells and formulas without using this function. The TRUE function is provided primarily for compatibility with other spreadsheet programs.

Date and Time

Dates are stored as sequential serial numbers so they can be used in calculations. By default, January 1, 1900 is serial number 1, and January 1, 2008 is serial number 39448 because it is 39,448 days after January 1, 1900.

Time values are a portion of a date value and represented by a decimal number (for example, 12:00 PM is represented as 0.5 because it is half of a day).

DATE	Returns the sequential serial number that represents a particular date.	*DATE(year,month,day)* Year- The year argument can be one to four digits. Month is a number representing the month of the year Day is a number representing the day of the month. If day is greater than the number of days in the month specified, day

		adds that number of days to the first day in the month. For example, DATE(2008,1,35) returns the serial number representing February 4, 2008.
DAY ***1**	Returns the day of a date, represented by a serial number. The day is given as an integer ranging from 1 to 31.	*DAY(serial_number)* Serial_number is the date of the day you are trying to find. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008.
HOUR	Returns the hour of a time value. The hour is given as an integer, ranging from 0 (12:00 A.M.) to 23 (11:00 P.M.).	*HOUR(serial_number)* Serial_number is the time that contains the hour you want to find. Times may be entered as text strings within quotation marks (for example, "6:45 PM"), as decimal numbers (for example, 0.78125, which represents 6:45 PM), or as results of other formulas or functions (for example, TIMEVALUE("6:45 PM")).
MINUTE	Returns the minutes of a time value. The minute is given as an integer, ranging from 0 to 59.	*MINUTE(serial_number)* Serial_number is the time that contains the minute you want to find. Times may be entered as text strings within quotation marks (for example, "6:45 PM"), as decimal numbers (for example, 0.78125, which represents 6:45 PM), or as results of other formulas or functions (for example, TIMEVALUE("6:45 PM")).
MONTH ***1**	Returns the month of a date represented by a serial number. The month is given as an integer, ranging from 1 (January) to 12 (December).	*MONTH(serial_number)* Serial_number is the date of the month you are trying to find. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008. Problems can occur if dates are entered as text
NOW	Returns the serial number of the current date and time.	*NOW( )* The NOW function changes only when the worksheet is calculated. It is not updated continuously.
SECOND	Returns the seconds of a time value. The second is given as an integer in the range 0 (zero) to 59.	*SECOND(serial_number)* Serial_number is the time that contains the seconds you want to find. Times may be entered as text strings within quotation marks (for example, "6:45 PM"), as decimal numbers (for example, 0.78125, which represents 6:45 PM), or as results of other formulas or functions (for example, TIMEVALUE("6:45 PM")). Time values are a portion of a date value and represented by a decimal number (for example, 12:00 PM is represented as 0.5 because it is half of a day).
TIME	Returns the decimal number for a particular time. The decimal number returned by TIME is a value ranging from 0	*TIME(hour,minute,second)* Hour is a number from 0 (zero) to 32767 representing the hour. Any value greater than 23 will be divided by 24 and the remainder will be treated as the hour value. For example,

	(zero) to 0.99999999, representing the times from 0:00:00 (12:00:00 AM) to 23:59:59 (11:59:59 P.M.).	TIME(27,0,0) = TIME(3,0,0) = .125 or 3:00 AM. Minute is a number from 0 to 32767 representing the minute. Any value greater than 59 will be converted to hours and minutes. For example, TIME(0,750,0) = TIME(12,30,0) = .520833 or 12:30 PM. Second is a number from 0 to 32767 representing the second. Any value greater than 59 will be converted to hours, minutes, and seconds. For example, TIME(0,0,2000) = TIME(0,33,22) = .023148 or 12:33:20 AM
TODAY	Returns the serial number of the current date. The serial number is the date-time code used for date and time calculations.	*TODAY( )*
WEEKDAY	Returns the day of the week corresponding to a date. The day is given as an integer, ranging from 1 (Sunday) to 7 (Saturday), by default.	*WEEKDAY(serial_number,return_type)* Serial_number is a sequential number that represents the date of the day you are trying to find. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008.
YEAR ***1**	Returns the year corresponding to a date. The year is returned as an integer in the range 1900-9999.	*YEAR(serial_number)* Serial_number is the date of the year you want to find. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008.

*1 - Values returned by the YEAR, MONTH and DAY functions will be Gregorian values regardless of the display format for the supplied date value. For example, if the display format of the supplied date is Hijri, the returned values for the YEAR, MONTH and DAY functions will be values associated with the equivalent Gregorian date.

Information

ERROR.TYPE *(see index of the different errors and their codes in the end of Information functions)*	Returns a number corresponding to one of the error values in Microsoft Excel or returns the #N/A error if no error exists. You can use ERROR.TYPE in an IF function to test for an error value and return a text string, such as a message, instead of the error value.	*ERROR.TYPE(error_val)* Error_val is the error value whose identifying number you want to find. Although error_val can be the actual error value, it will usually be a reference to a cell containing a formula that you want to test.

ISBLANK *([see IS Functions](#))*	Checks the type of value and returns TRUE or FALSE depending on the outcome. Returns TRUE if the value is blank.	*ISBLANK(value)* Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.
ISERR *([see IS Functions](#))*	Checks the type of value and returns TRUE or FALSE depending on the outcome. Returns TRUE if the value is any error value except #N/A.	*ISERR(value)* Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.
ISERROR *([see IS Functions](#))*	Checks the type of value and returns TRUE or FALSE depending on the outcome. Returns TRUE if the value is any an error value.	*ISERROR(value)* Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.
ISLOGICAL *([see IS Functions](#))*	Checks the type of value and returns TRUE or FALSE depending on the outcome. Returns TRUE if the value is a logical value.	*ISLOGICAL(value)* Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.
ISNA *([see IS Functions](#))*	Checks the type of value and returns TRUE or FALSE depending on the outcome. Returns TRUE if the value is the #N/A error value.	*ISNA(value)* Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.
ISNONTEXT *([see IS Functions](#))*	Checks the type of value and returns TRUE or FALSE depending on the outcome. Returns TRUE if the value is not text.	*ISNONTEXT(value)* Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.
ISNUMBER *([see IS Functions](#))*	Checks the type of value and returns TRUE or FALSE depending on the outcome. Returns TRUE if the value is a number	*ISNUMBER(value)* Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.
ISREF *([see IS Functions](#))*	Checks the type of value and returns TRUE or FALSE depending on the outcome. Returns TRUE if the value is a reference.	*ISREF(value)* Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.
ISTEXT *([see IS Functions](#))*	Checks the type of value and returns TRUE or FALSE depending on the outcome. Returns TRUE if the value is text.	*ISTEXT(value)* Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.

ISEVEN	Returns TRUE if number is even, or FALSE if number is odd. If this function is not available, and returns the #NAME? Error.	*ISEVEN(number)* Number is the value to test. If number is not an integer, it is truncated.
ISODD	Returns TRUE if number is odd, or FALSE if number is even. If this function is not available, and returns the #NAME? Error.	*ISODD(number)* Number is the value to test. If number is not an integer, it is truncated. If number is nonnumeric, ISODD returns the #VALUE! error value.
N (**see N function values**)	Returns a value converted to a number.	*N(value)* Value is the value you want converted. N converts values listed in the following table.
NA	Returns the error value #N/A. #N/A is the error value that means "no value is available." Use NA to mark empty cells. By entering #N/A in cells where you are missing information, you can avoid the problem of unintentionally including empty cells in your calculations.	*NA( )*

ERROR TYPES

If error_val is	ERROR.TYPE returns
#NULL!	1
#DIV/0!	2
#VALUE!	3
#REF!	4
#NAME?	5
#NUM!	6
#N/A	7
Anything else	#N/A

IS FUNCTIONS

Each of these functions, referred to collectively as the IS functions, checks the type of value and returns TRUE or FALSE depending on the outcome. For example, the ISBLANK function returns the logical value TRUE if value is a reference to an empty cell; otherwise it returns FALSE.

ISBLANK(value), ISERR(value), ISERROR(value), ISLOGICAL(value), ISNA(value), ISNONTEXT(value), ISNUMBER(value), ISREF(value), ISTEXT(value)

The IS functions are useful in formulas for testing the outcome of a calculation. When combined with the IF function, they provide a method for locating errors in formulas (see the following examples).

Function	Returns TRUE if
ISBLANK	Value refers to an empty cell.
ISERR	Value refers to any error value except #N/A.
ISERROR	Value refers to any error value (#N/A, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!).
ISLOGICAL	Value refers to a logical value.
ISNA	Value refers to the #N/A (value not available) error value.
ISNONTEXT	Value refers to any item that is not text. (Note that this function returns TRUE if value refers to a blank cell.)
ISNUMBER	Value refers to a number.
ISREF	Value refers to a reference.
ISTEXT	Value refers to text.

N FUNCTION VALUES

If value is or refers to	N returns
A number	That number
A date, in one of the built-in date formats available in Microsoft Excel	The serial number of that date
TRUE	1
FALSE	0
An error value, such as #DIV/0!	The error value

Lookup and Reference

CHOOSE	Uses index_num to return a value from the list of value arguments. Use CHOOSE to select one of up to 29 values based on the index number. For example, if value1 through value7 are the days of the week, CHOOSE returns one of the days when a number between 1 and 7 is used as index_num.	*CHOOSE(index_num,value1,value2,...)* Index_num specifies which value argument is selected. Index_num must be a number between 1 and 29, or a formula or reference to a cell containing a number between 1 and 29. Value1,value2,... are 1 to 29 value arguments from which CHOOSE selects a value or an action to perform based on index_num. The arguments can be numbers, cell references, defined names, formulas, functions, or text.
COLUMN	Returns the column number of the given reference.	*COLUMN(reference)* Reference is the cell or range of cells for which you want the column number. If reference is omitted, it is assumed to be the reference of the cell in which the COLUMN function appears. Reference cannot refer to multiple areas.
COLUMNS	Returns the number of columns in an array or reference.	*COLUMNS(array)* Array is an array or array formula, or a reference to a range of cells for which you want the number of columns.
HLOOKUP	Searches for a value in the top row of an array of values, and then returns a value in the same column from a row you specify in the table or array.	*HLOOKUP(lookup_value,table_array,row_index_num,range_lookup)* Lookup_value is the value to be found in the first row of the array. Lookup_value can be a value, a reference, or a text string. Table_array is a table of information in which data is looked up. Use a reference to a range or a range name. Row_index_num is the row number in table_array from which the matching value will be returned. A row_index_num of 1 returns the first row value in table_array, a row_index_num of 2 returns the second row value in table_array, and so on. Range_lookup is a logical value that specifies whether you want HLOOKUP to find an exact match or an approximate match. If TRUE or omitted, an approximate match is returned. In other words, if an exact match is not found, the next largest value that is less than

		lookup_value is returned.
INDEX	Returns the value of an element in a table or an array, selected by the row and column number indexes. The INDEX function has two syntax forms: array and reference. The array form always returns a value or array of values; the reference form always returns a reference. Use the array form if the first argument to INDEX is an array constant.	*INDEX(array,row_num,column_num)* Array is a range of cells or an array constant. If array contains only one row or column, the corresponding row_num or column_num argument is optional. If array has more than one row and more than one column, and only row_num or column_num is used, INDEX returns an array of the entire row or column in array. Row_num selects the row in array from which to return a value. If row_num is omitted, column_num is required. Column_num selects the column in array from which to return a value. If column_num is omitted, row_num is required. *INDEX(reference,row_num,column_num,area_num)* Reference is a reference to one or more cell ranges. If you are entering a nonadjacent range for the reference, enclose reference in parentheses. If each area in reference contains only one row or column, the row_num or column_num argument, respectively, is optional. For example, for a single row reference, use INDEX(reference,,column_num). Row_num is the number of the row in reference from which to return a reference. Column_num is the number of the column in reference from which to return a reference. Area_num selects a range in reference from which to return the intersection of row_num and column_num. The first area selected or entered is numbered 1, the second is 2, and so on. If area_num is omitted, INDEX uses area 1.
LOOKUP	Returns a value either from a one-row or one-column range or from an array. The LOOKUP function has two syntax forms: vector and array. The vector form of LOOKUP looks in a one-row or one-column range (known as a vector) for a value and returns a value from the same position in a second one-row or one-column range. The array form of LOOKUP looks in the first row or column of an array for the	*LOOKUP(lookup_value,lookup_vector,result_vector)* Lookup_value is a value that LOOKUP searches for in the first vector. Lookup_value can be a number, text, a logical value, or a name or reference that refers to a value. Lookup_vector is a range that contains only one row or one column. The values in lookup_vector can be text, numbers, or logical values. The values in lookup_vector must be placed in ascending order: ...,-2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE; otherwise, LOOKUP may not give the correct value. Uppercase and lowercase text are equivalent. Result_vector is a range that contains only one row or column. It must be the same size as lookup_vector.

	specified value and returns a value from the same position in the last row or column of the array.	*LOOKUP(lookup_value,array)* Lookup_value is a value that LOOKUP searches for in an array. Lookup_value can be a number, text, a logical value, or a name or reference that refers to a value. If LOOKUP can't find the lookup_value, it uses the largest value in the array that is less than or equal to lookup_value. If lookup_value is smaller than the smallest value in the first row or column (depending on the array dimensions), LOOKUP returns the #N/A error value. Array is a range of cells that contains text, numbers, or logical values that you want to compare with lookup_value. The array form of LOOKUP is very similar to the HLOOKUP and VLOOKUP functions. The difference is that HLOOKUP searches for lookup_value in the first row, VLOOKUP searches in the first column, and LOOKUP searches according to the dimensions of array. If array covers an area that is wider than it is tall (more columns than rows), LOOKUP searches for lookup_value in the first row. If array is square or is taller than it is wide (more rows than columns), LOOKUP searches in the first column. With HLOOKUP and VLOOKUP, you can index down or across, but LOOKUP always selects the last value in the row or column. The values in array must be placed in ascending order: ...,-2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE; otherwise, LOOKUP may not give the correct value. Uppercase and lowercase text are equivalent.
MATCH	Returns the relative position of an item in an array or a reference that matches a specified value in a specified order. Use MATCH instead of one of the LOOKUP functions when you need the position of an item in a range instead of the item itself.	*MATCH(lookup_value,lookup_array,match_type)* Lookup_value is the value you use to find the value you want in a table. Lookup_value is the value you want to match in lookup_array. For example, when you look up someone's number in a telephone book, you are using the person's name as the lookup value, but the telephone number is the value you want. Lookup_value can be a value (number, text, or logical value) or a cell reference to a number, text, or logical value. Lookup_array is a contiguous range of cells containing possible lookup values. Lookup_array must be an array or an array reference. Match_type is the number -1, 0, or 1. Match_type specifies how Microsoft Excel matches lookup_value with values in lookup_array.

		If match_type is 1, MATCH finds the largest value that is less than or equal to lookup_value. Lookup_array must be placed in ascending order: ...-2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE.
		If match_type is 0, MATCH finds the first value that is exactly equal to lookup_value. Lookup_array can be in any order.
		If match_type is -1, MATCH finds the smallest value that is greater than or equal to lookup_value. Lookup_array must be placed in descending order: TRUE, FALSE, Z-A, ...2, 1, 0, -1, -2, ..., and so on.
		If match_type is omitted, it is assumed to be 1.
ROW	Returns the row number of a reference.	*ROW(reference)* Reference is the cell or range of cells for which you want the row number. If reference is omitted, it is assumed to be the reference of the cell in which the ROW function appears. Reference cannot refer to multiple areas.
ROWS	Returns the number of rows in a reference or array.	*ROWS(array)* Array is an array, an array formula, or a reference to a range of cells for which you want the number of rows.
TRANSPOSE	Returns a vertical range of cells as a horizontal range, or vice versa. TRANSPOSE must be entered as an array formula in a range that has the same number of rows and columns, respectively, as an array has columns and rows. Use TRANSPOSE to shift the vertical and horizontal orientation of an array on a worksheet.	*TRANSPOSE(array)* Array is an array or range of cells on a worksheet that you want to transpose. The transpose of an array is created by using the first row of the array as the first column of the new array, the second row of the array as the second column of the new array, and so on.
VLOOKUP	Searches for a value in the leftmost column of a table, and then returns a value in the same row from a column you specify in the table. Use VLOOKUP instead of HLOOKUP when your comparison values are located in a column to the left of the data you want to find. The V in VLOOKUP stands for "Vertical."	*VLOOKUP(lookup_value,table_array,col_index_num,range_lookup)* Lookup_value is the value to be found in the first column of the array. Lookup_value can be a value, a reference, or a text string. Table_array is the table of information in which data is looked up. Use a reference to a range or a range name, such as Database or List. If range_lookup is TRUE, the values in the first column of table_array must be placed in ascending order: ..., -2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE; otherwise VLOOKUP may not give the correct value. If range_lookup is FALSE, table_array

| | | stands for "Vertical." | does not need to be sorted.

You can put the values in ascending order by choosing the Sort command from the Data menu and selecting Ascending.

The values in the first column of table_array can be text, numbers, or logical values.

Col_index_num is the column number in table_array from which the matching value must be returned. A col_index_num of 1 returns the value in the first column in table_array; a col_index_num of 2 returns the value in the second column in table_array, and so on

Range_lookup is a logical value that specifies whether you want VLOOKUP to find an exact match or an approximate match. If TRUE or omitted, an approximate match is returned. In other words, if an exact match is not found, the next largest value that is less than lookup_value is returned |

Text and Data

CHAR	Returns the character specified by a number. Use CHAR to translate code page numbers you might get from files on other types of computers into characters.	*CHAR(number)* Number is a number between 1 and 255 specifying which character you want. The character is from the character set used by your handheld.
CONCAT ENATE	Joins several text strings into one text string.	*CONCATENATE (text1,text2,...)* Text1, text2, ... are 1 to 30 text items to be joined into a single text item. The text items can be text strings, numbers, or single-cell references.
EXACT	Compares two text strings and returns TRUE if they are exactly the same, FALSE otherwise. EXACT is case-sensitive but ignores formatting differences. Use EXACT to test text being entered into a document.	*EXACT(text1,text2)* Text1 is the first text string. Text2 is the second text string.
FIND	FIND finds one text string (find_text) within another text string (within_text), and returns the number of the starting position of find_text, from the first character of within_text. You can also use SEARCH to find one text string within another, but unlike SEARCH, FIND is	*FIND(find_text,within_text,start_num)* Find_text is the text you want to find. Within_text is the text containing the text you want to find. Start_num specifies the character at which to start the search. The first character in within_text is character number 1. If you omit start_num, it is assumed to be 1.

		case sensitive and doesn't allow wildcard characters.	
LEFT	LEFT returns the first character or characters in a text string, based on the number of characters you specify.	*LEFT(text,num_chars)* Text is the text string that contains the characters you want to extract. Num_chars specifies the number of characters you want LEFT to extract. Num_chars must be greater than or equal to zero. If num_chars is greater than the length of text, LEFT returns all of text. If num_chars is omitted, it is assumed to be 1.
LEN	LEN returns the number of characters in a text string.	*LEN(text)* Text is the text whose length you want to find. Spaces count as characters.
LOWER	Converts all uppercase letters in a text string to lowercase.	*LOWER(text)* Text is the text you want to convert to lowercase. LOWER does not change characters in text that are not letters.
MID	MID returns a specific number of characters from a text string, starting at the position you specify, based on the number of characters you specify.	*MID(text,start_num,num_chars)* Text is the text string containing the characters you want to extract. Start_num is the position of the first character you want to extract in text. The first character in text has start_num 1, and so on. Num_chars specifies the number of characters you want MID to return from text.
PROPER	Capitalizes the first letter in a text string and any other letters in text that follow any character other than a letter. Converts all other letters to lowercase letters.	*PROPER(text)* Text is text enclosed in quotation marks, a formula that returns text, or a reference to a cell containing the text you want to partially capitalize.
REPLACE	REPLACE replaces part of a text string, based on the number of characters you specify, with a different text string.	*REPLACE(old_text,start_num,num_chars,new_text)* Old_text is text in which you want to replace some characters. Start_num is the position of the character in old_text that you want to replace with new_text. Num_chars is the number of characters in old_text that you want REPLACE to replace with new_text. New_text is the text that will replace characters in old_text.
REPT	Repeats text a given number of times. Use REPT to fill a cell with a number of instances of a	*REPT(text,number_times)* Text is the text you want to repeat.

	text string. If number_times is 0 (zero), REPT returns "" (empty text). If number_times is not an integer, it is truncated.	Number_times is a positive number specifying the number of times to repeat text.
RIGHT	RIGHT returns the last character or characters in a text string, based on the number of characters you specify. Num_chars must be greater than or equal to zero. If num_chars is greater than the length of text, RIGHT returns all of text. If num_chars is omitted, it is assumed to be 1.	*RIGHT(text,num_chars)* Text is the text string containing the characters you want to extract. Num_chars specifies the number of characters you want RIGHT to extract. Num_bytes specifies the number of characters you want RIGHTB to extract, based on bytes.
SUBSTITUTE	Substitutes new_text for old_text in a text string. Use SUBSTITUTE when you want to replace specific text in a text string; use REPLACE when you want to replace any text that occurs in a specific location in a text string.	*SUBSTITUTE(text,old_text,new_text,instance_num)* Text is the text or the reference to a cell containing text for which you want to substitute characters. Old_text is the text you want to replace. New_text is the text you want to replace old_text with. Instance_num specifies which occurrence of old_text you want to replace with new_text. If you specify instance_num, only that instance of old_text is replaced. Otherwise, every occurrence of old_text in text is changed to new_text.
T	Returns the text referred to by value. If value is or refers to text, T returns value. If value does not refer to text, T returns "" (empty text). You do not generally need to use the T function in a formula because Microsoft Excel automatically converts values as necessary. This function is provided for compatibility with other spreadsheet programs.	*T(value)* Value is the value you want to test.
TRIM	Removes all spaces from text except for single spaces between words. Use TRIM on text that you have received from	*TRIM(text)* Text is the text from which you want spaces removed.

	another application that may have irregular spacing.		53
UPPER	Converts text to uppercase.	*UPPER(text)* Text is the text you want converted to uppercase. Text can be a reference or text string.	

Synchronization

Mobile Excel conduit

Mobile Excel Conduit is available for Windows PC and allows you to transfer Microsoft Excel workbooks to Mobile Excel and the other way around. The transfer of Excel files is through:

- o XML files for Microsoft Excel XP and 2003
- o CSV files for older versions of Microsoft Excel (e.g. 2000, 98)

To make it easier for you to synchronize, Mobile Excel conduit will automatically detect the version of Microsoft Office that you use and create the necessary XML or CSV.

If you synchronize through CSV files, make sure that you have configured Mobile Excel correctly. If you have not specified the type of the documents that you import, Mobile Excel will not be capable of opening imported Microsoft Excel files.

To set Mobile Excel to handle Microsoft Excel 98 or 2000 files:

1. Start the program on your Palm device. The *workbook launcher* will open.
2. Open the main menu and select **Options**> **Preferences**.
3. Set the correct **CSV Separator.** The CSV Separator should be the same as the **List Separator** in the *Regional Options* in the *Control Panel* on your Windows desktop PC.
4. Tap OK to close the Preferences form.

How to open Mobile Excel conduit preferences?

Mobile Excel conduit is installed during the initial installation of the program. To access the conduit settings window:

1. Right-click on the HotSync icon in the system tray of your desktop computer.
2. Select **Custom** from the popup menu that will appear.
3. Scroll in the conduit list until you locate Mobile Excel 2004 conduit.
4. Select the conduit and click the **Change** button.
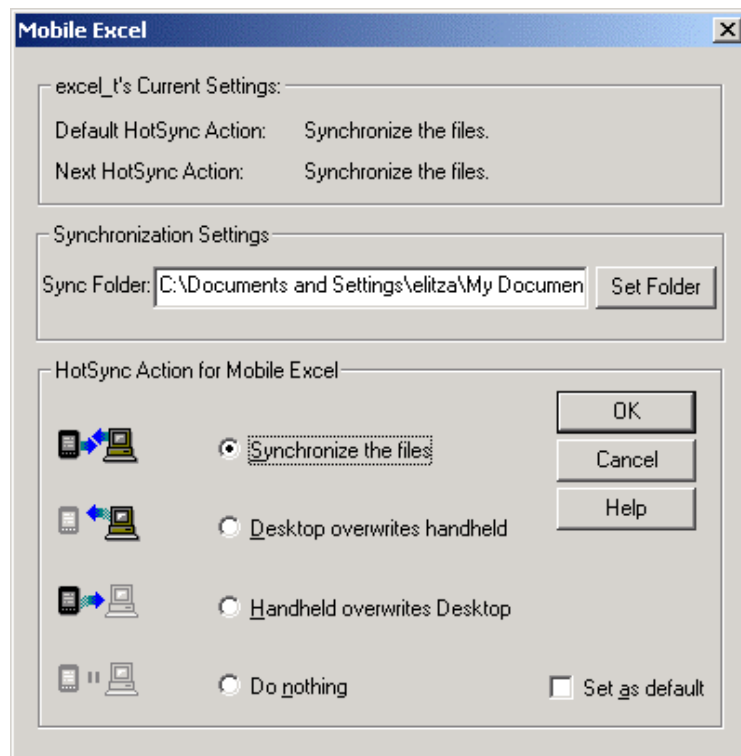
Synchronization Folder

Mobile Excel conduit will transfer to the Palm device only Microsoft Excel spreadsheet files. The default synchronization directory is **\My Documents\Mobile Excel** and is created during the installation of the program. All files that you want to import to the Palm edition of Mobile Excel should be copied to that directory.

Mobile Excel conduit gives you the opportunity to change the default synchronization folder according to your preferences. To change the default directory:

1. Open Mobile Excel conduit settings as described in the previous chapter.

2. Click the **Set Folder** button.

3. Browse the directory structure on your desktop computer and select the new folder that you want to use for synchronization.

4. Click OK to confirm the directory selection and close the *Browse for Folder* window.

5. If you want to use the selected folder when you synchronize from now on, enable the **Set as default** check box in Mobile Excel Conduit window.

Please note that when you change the default directory, the files from the old synchronization folder will not be copied to the new one. If you want to continue to synchronize these files, you have to copy these files manually to the new default folder.
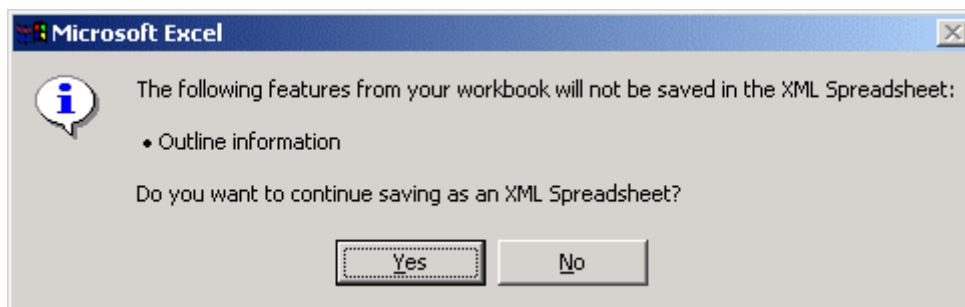

Synchronization with Excel XP or 2003

When you transfer Microsoft Excel workbook to Mobile Excel the actual transfer is through the XML Spreadsheet format. Due to the specifics of the XML format the following types of objects and data are not synchronized with the file:

- o OLE objects (such as embedded documents or objects from another program)
- o Pictures
- o Drawings and AutoShapes
- o Charts
- o Microsoft Visual Basic for Applications (VBA) projects
- o Group and outline information
- o Custom e-mail envelope information
- o Custom named styles that are not being used by any cell in the workbook

You don't have to save your MS Excel XLS files as XMLs. It is necessary to just copy the XLS files to the default synchronization directory and Mobile Excel conduit will create temporary XML files and import them in the Palm application.

For each synchronized Microsoft Excel file which contains any of these objects, Mobile Excel conduit will show the following warning massage, notifying you that the objects will be lost:

Click **Yes** to transfer the workbook anyway, or No to cancel the document import for this particular file.

## *Synchronization with Excel 2000*

When you workbooks created in Microsoft Excel 2000 are exported to Mobile Excel the actual transfer is through the frequently used CSV (comma separated values) file format. In addition to exporting the CSV file to the Palm application, Mobile Excel will convert the standard XLS Excel file to a CSV one so you don't have to save

manually your worksheet as CSVs. All you have to do in order to export Microsoft Excel 2000 file to Mobile Excel is to copy the file to the *default synchronization folder* used by Mobile Excel.

However, the CSV format does not support all object types allowed in Microsoft Excel and after such files are uploaded on the Palm device the following types of objects from the original XLS file will be lost:

- o OLE objects (such as embedded documents or objects from another program)
- o Pictures
- o Drawings and AutoShapes
- o Charts
- o Microsoft Visual Basic for Applications (VBA) projects
- o Group and outline information
- o Custom e-mail envelope information
- o Custom named styles that are not being used by any cell in the workbook

For each MS Excel 2000 file which is synchronized and contains any of the objects listed above, Mobile Excel conduit will show the following warning during HotSync operations:
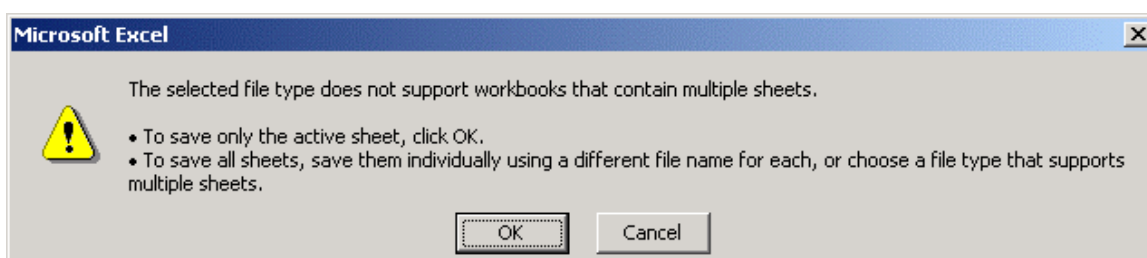
Click **OK** – to export the file to the handheld, or **Cancel** – to stop the transfer.

Each CSV file can contain data from just one spreadsheet. This is why when you export Microsoft Excel 2000 XLS file to Mobile Excel, only the first spreadsheet of each synchronized workbook will be transferred to the handheld device.

Synchronizing with Excel 97

The export of Microsoft Excel 97 files to Mobile Excel is very similar to the synchronization of Excel 2000 files described in the previous chapter. In the same way the transfer is through comma separated value (CSV) files. However, Mobile Excel

conduit is not capable to create the CSV file from the original Microsoft Excel XLS file.

To be able to export Microsoft Excel 97 workbooks to Mobile Excel, you should save the workbook as a CSV file in Mobile Excel beforehand!

Each CSV file can contain a single worksheet but not a whole Excel workbook.

To save an Excel worksheet as a CSV:

1. Open Microsoft Excel and the file you wish to export.

2. Open the main menu and select **File** > **Save as.**

3. Type in a name for the file in the **File Name** field.

4. Select **CSV (Comma delimited)(*.csv)** from the popup list for the **Save as type** field.

5. The following warning message will appear before the file is saved.
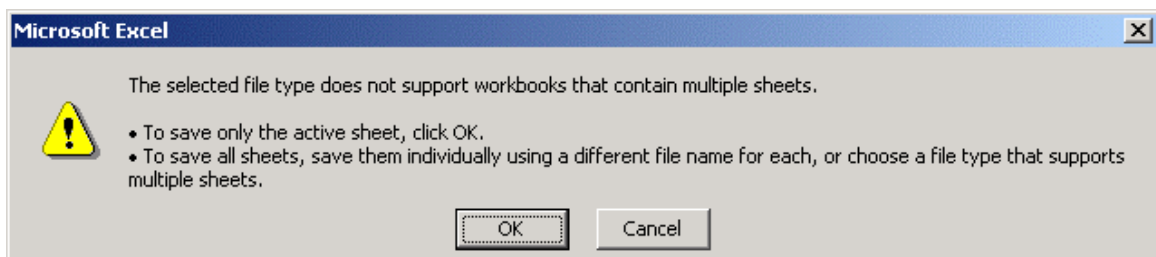
6. Click OK to save the worksheet in CSV format.

The only thing that you should do before you are ready to synchronize and upload the worksheet to Mobile Excel is to copy the created CSV file to the *default synchronization folder*.

Synchronization directions

Mobile Excel conduit allows you to select direction in which the files will be transferred during HotSync operation. The four possible Mobile Excel actions are described below.

To set any of the actions as a default one, just open the Mobile Excel conduit settings, select the action, and enable the check box for **Set as default.**

Desktop Overwrites Handheld

All files from the *default synchronization folder,* set in the Mobile Excel conduit settings on your desktop, are exported to the Palm device.

> All workbooks with the same names as files from the synchronization folder will overwrite the documents on the handheld.

Handheld Overwrites Desktop

All workbooks from Mobile Excel on the Palm device will be exported to the *default synchronization folder* used by Mobile Excel. The files will be transferred as XLS files and can be directly opened by Microsoft Excel.

> If the synchronization folder already contains files with the same file names as workbooks on the Palm device, the Palm documents will overwrite the ones on the desktop PC.

Synchronize the files

To avoid overwriting of data *synchronize the files* will transfer only new workbooks from the default desktop synchronization directory to the Palm device and the other way around, only Palm workbooks that do not exist in the synchronization directory will be transferred on the desktop.

> Only new workbooks created either on the Palm device, or added to the synchronization folder on the desktop PC are synchronized during the *Synchronize the files* transfer.

Do Nothing

This action will disable Mobile Excel conduit and files will not be transferred in neither direction.