element k JOURNALS
A division of Eli Research

Inside Microsoft Excel

Volume 12, Number 12 • December 2005 • www.elementkjournals.com

In this issue:

Editor's Note:

Relax—the holiday season is almost over and soon you'll be ringing in a New Year! End the old year off right by checking out the how-to techniques I've included in this month's issue. Frank Kuhn leads you through some sticky investment situations by showing you how Excel can help maximize your returns while minimizing your risk. I'll demonstrate four user-defined functions that translate numerical currency values into English words to aid your accounting needs. Please feel free to contact me, even if it's just to introduce yourself. I look forward to hearing from you.

Kara

ime_editor@elementk.com
Toll-free: (866)614-8715

Analysis Tools

Are your investments too risky? Find out with the Solver tool

by Frank Kuhn

AT A GLANCE

Knowing which stocks to pick and in what quantities so as to maximize your returns while minimizing your risk is a trick that can baffle even the best of the Wall Street wizards. You can take a giant step toward mastering this trick—even if you don't have years of experience or education under your belt—by mastering Excel's Solver tool.

To show you how to maximize investment returns while minimizing investment risk, we'll:

- Set up a worksheet that illustrates the problem facing all investors: how to increase your gains without accruing too much risk.
- Point out the various constraints and variables present in such a problem.
- Introduce you to the Solver tool and use it to unravel this investment riddle.

Application: Microsoft Excel 2000/2002/2003/2004

Operating Systems: Microsoft Windows, Macintosh

Keywords: Excel, Financial Analysis, Diversification, Scenario-Testing, Investment Risk, Solver Tool

Download: http://download.elementkjournals.com/excel/200512/solver.zip

Investors want to maximize the overall return while minimizing the overall risk, but they only have so much money to work with. The difficulty lies in the number and complexity of the related variables, constraints, and conditions involved. And, each stock in the portfolio has its own beta risk measure as well as its own expected return. Goal Seek can only handle scenario-testing dealing with one constraint. Luckily, here's where the Solver tool comes to the rescue.

Set up a portfolio worksheet

Most investors have not just one but several investments in their portfolio. Each of these investments has its own respective degree of risk and expected return. Some of these investments offer greater returns than others, but they also likely bring greater risk in the bargain. How do you adjust the weights of these investments so as to optimize and balance both risk and reward?

To illustrate this problem, we've set up a sample portfolio spreadsheet, as shown in **Figure A**. In our sample portfolio, we have three stocks: Stock A, B, and C. You're investigating Stock X, whose beta is 1.39. It has an expected return of 15.6 percent. You're interested in this exciting but somewhat risky stock, but you don't want your portfolio beta to exceed the market beta of 1.0: That is to say, you don't want your portfolio to be any more risky than the market as a whole. Using risk-free T-bills as a counter-weight to the risky Stock X, how much of Stock X in relation to T-bills should you buy to maximize your portfolio return while maintaining a portfolio beta of 1.0?

Note: You can follow along with our example by downloading and extracting the file **investment.xls** from the URL given at the beginning of this article or build your own worksheet using ours as a guide.

To answer this question, we'll use Excel's Solver tool. Before explaining how to use it, however, we'll first show you how to set up this portfolio worksheet.

To set up a portfolio worksheet:

1. Enter the necessary column headings and formatting, using **Figure A** as your guide.
2. Apply Percentage formatting to columns C, E, and G, if desired.
3. Enter your security names, the amount invested in each, and their respective returns and betas. Or, use our data to follow along with our example.
4. Enter `=SUM(B2:B6)` in cell B7.

Tip: Don't forget that you can create `SUM` formulas in a flash with the AutoSum button Σ -, which you can find on the Standard toolbar. Just select a cell, click the AutoSum button, ensure that it correctly selects the cells you want to sum, and then press [Enter].

5. Type `=B2/$B$7` in cell E2 to determine Stock A's weight in the portfolio. Press [Enter].
6. Enter `=D2*E2` in cell F2 to determine Stock A's portfolio beta (that stock's beta as modified by its portfolio weight). Press [Enter].
7. Enter `=C2*E2` in cell G2 to determine Stock A's portfolio return (its return as modified by its portfolio weight). Press [Enter].
8. Select cells E2:G2 and, using the cells' Fill handle, drag to cell G6. The weights, portfolio betas and returns should appear in those cells.

Now you only need to enter the formulas to determine your total weights, portfolio beta, and return.

To determine your total portfolio weights, beta, and return:

1. Enter `=SUM(E2:E6)` in cell E7. The total portfolio weight of 100% appears.
2. Click and drag cell E7's Fill handle over to cell G7. The total portfolio beta and return should also appear.

Now you're ready to use the Solver tool to find what quantities of Stock X and T-bills to buy in order to reduce your beta while maximizing your return.

Note: To copy cells without formatting (versions 2002/2003), just click on the Smart Tag that appears after you copy the cells and choose Fill Without Formatting.

Utilize the Solver tool

The Solver tool belongs to Excel's suite of what-if analysis tools. Solver, unlike Goal Seek, can find an optimal spreadsheet value (the target cell), by adjusting several related formula cells while subjecting the model to a number of restrictions or constraints. Let's first take a closer look at the Solver Parameters dialog box.

Take a tour of the Solver Parameters dialog box

To open the Solver tool (once the add-in is installed), just select Tools | Solver from the menu bar. The Solver Parameters dialog box appears, as shown in **Figure B**.

The Solver Parameters dialog box is divided into three main sections:

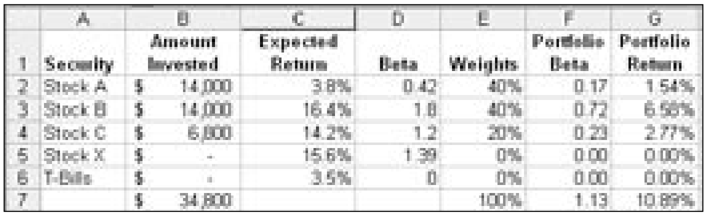
- **Set Target Cells.** This cell is the target or objective cell that you either want to maximize, minimize, or set to a specific value. You can specify this by selecting one of the Max, Min, or Value Of option buttons, respec-

Related Articles

For a closer look at T-bills and investment betas, check out these articles:

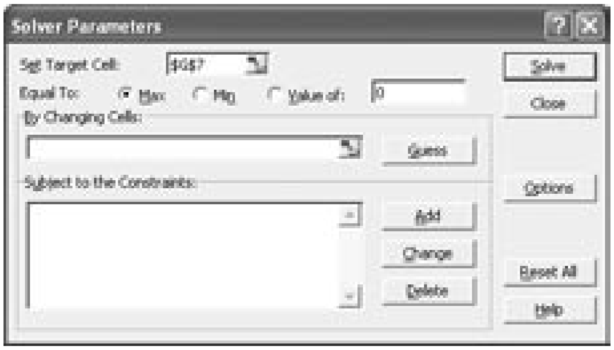
- Take the guesswork out of your investing with the Regression Analysis tool (November 2005)
- Make better "cents" of U.S. Treasury bills using Excel's T-bill functions (October 2005)
- Easily answer what-if questions using Excel's Goal Seek feature (October 2005)

These articles and more are available to EKJ Online subscribers at **www.elementkjournals.com/premier/journal.asp**.



	A	B	C	D	E	F	G
1	Security	Amount Invested	Expected Return	Beta	Weights	Portfolio Beta	Portfolio Return
2	Stock A	$ 14,000	3.8%	0.42	40%	0.17	1.54%
3	Stock B	$ 14,000	16.4%	1.8	40%	0.72	6.56%
4	Stock C	$ 6,800	14.2%	1.2	20%	0.23	2.77%
5	Stock X	$ -	15.6%	1.39	0%	0.00	0.00%
6	T-Bills	$ -	3.5%	0	0%	0.00	0.00%
7		$ 34,800			100%	1.13	10.89%

A: Our portfolio spreadsheet takes into account both risk and reward.



Solver Parameters

Set Target Cell: G7

Equal To: ● Max ○ Min ○ Value of: 0

By Changing Cells:

Subject to the Constraints:

B: The Solver Parameters dialog box allows you greater range and flexibility in what-if scenario testing than Goal Seek.

tively, in the Equal To area. The target cell must contain a formula.

- **By Changing Cells.** These cells are the ones you'll be changing to reach your target in the Target Cell. Use a comma to separate multiple changing cells.
- **Subject To The Constraints.** In these cells you'll set constraints on your changing cells.

After you've entered data in your target cell and its goal, your changing cells, and constraints, click the Solve button to find your optimal solution. Or, click the Close button to close the Solver tool without giving a solution.

For advanced options: Click the Options button to access the Solver Options dialog box, which allows you to set the max time, iterations, precision, tolerance, and/or convergence of your optimization modeling. You also have the option to save or load saved optimization scenarios, as well as set other conditions for advanced modeling.

Now that you're familiar with the Solver Parameters dialog box, let's apply it to our investment conundrum.

Find your optimal portfolio investment mix

Looking over your portfolio spreadsheet, you see that cell G7, 10.89%, your total portfolio return, is the cell you want to maximize. Cell F7, 1.13, is your current portfolio beta. You'd like to reduce this to 1.0 so your portfolio is no riskier than the overall market. This will be one of your constraints. Cell B7 contains your total investment so far, $34,800: you only have another $15,200 left to invest. This means that the total investment ($50,000) is another constraint. Your final constraint is Stock X itself; you'd like to buy at least some of this stock, so we'll set this as greater than or equal to one.

The cells that you'll be changing, of course, are B5 and B6, the amount invested in Stock X and T-bills. All other cells, your previous investments, should remain unchanged. Only the cells you designate in the Solver Parameter dialog box's Set Target Cell text box, By Changing Cells text box, and Subject To The Constraints list box will change.

To find your optimal investment mix using Solver:

1. Select cell G7, your total portfolio return, to enter it in the Set Target Cell text box.
2. Choose the Max option button if it isn't already selected.
3. Click in the By Changing Cells text box to activate it, select cell B5, enter a comma (,) and then select B6 to make the Stock X and T-bill investments your changing cells.
4. Click in the Subject To The Constraints list box to activate it and click the Add button to add your first constraint. The Add Constraint dialog box opens, as shown in **Figure C**.
5. Refer to **Table A** to add three constraints, clicking the Add button after each one. When you're done, click OK instead of Add to return to the Solver Parameters dialog box.

Tip: You can click the Guess button in the Solver Parameters dialog box if you'd like Solver to automatically place the references of all nonformula cells referred to in the formula in the Set Target Cell box, as well as their references, in the By Changing Cells text box.

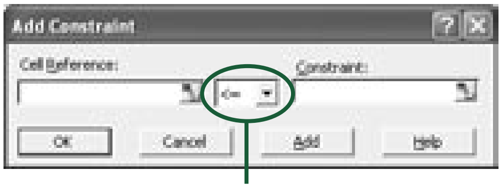
6. Upon clicking OK, the Solver Parameters dialog box appears, as shown in **Figure D** if you're following along with our example.

Install the Solver add-in

If you haven't already done so, you need to install the Solver tool onto your system via Excel's Add-in utility.

To load the Solver tool:

1. Open a new workbook in Excel.
2. Choose Tools | Add-Ins from the menu bar.
3. In the Add-Ins dialog box, select the Solver Add-in check box in the Add-Ins Available list box, and then click OK.



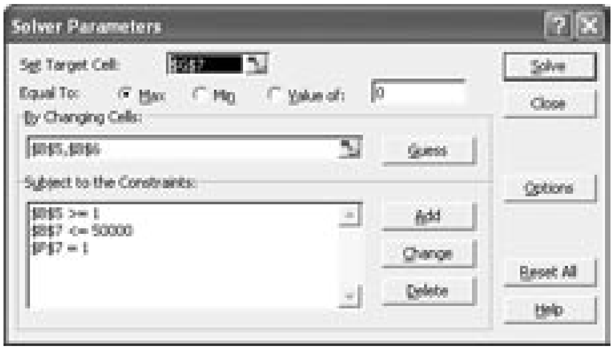
<=

Constraint relationship

C: Use the Add Constraint dialog box to add up to 100 constraints to your optimization model.

Table A: Solver tool constraints

Cell reference	Constraint relationship	Constraint
B5	>=	1
B7	<=	50000
F7	=	1.0



Set Target Cell: G7

Equal To: ● Max ○ Min ○ Value of: 0

By Changing Cells: B5,B6

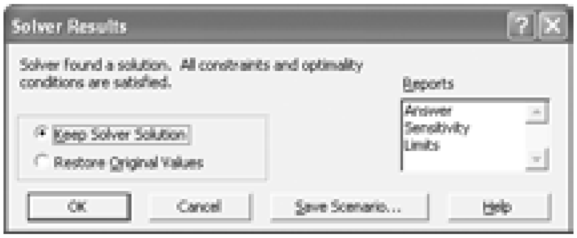
Subject to the Constraints:

B5 >= 1
B7 <= 50000
F7 = 1

D: Our optimization model has one target cell, two changing cells, and three constraints.

7. Click the Solve button to find the optimal investment mix solution. The Solver Results dialog box first appears, as shown in **Figure E**. Make sure the Keep Solver Solution option button is selected and click OK.

The optimal solution of 10.52% appears in G7. You can buy up to $7,741 in Stock X and $7,459 in T-bills and other risk-free assets without spending more than $50,000 while maintaining a portfolio beta of 1.0.


Solver found a solution. All constraints and optimality conditions are satisfied.

● Keep Solver Solution

○ Restore Original Values

E: Your results show up in a separate dialog box so you can decide whether you want to keep Excel's suggested solution.

Alternative uses for the Solver tool

In addition to finding optimal investment mixes, the Solver tool can handle nearly any optimization modeling that has multiple variables or constraints. For example, you might determine a company's ideal product mix, price, or place of production. You may even need to plan a project. In fact, the list of possible uses is nearly endless and includes any situation where you're interested in finding the most efficient means of achieving a given objective. ✳

Excel Enhancements

Four things Diff Doc can do that the Compare Side By Side feature can't

by Kara Soos

AT A GLANCE

Excel has increased its capability for multi-workbook analysis with Excel 2003's Compare Side By Side feature, but there are several reasons why it just isn't enough to adequately compare two workbooks. Diff Doc is a utility that takes document comparison to a whole new level, giving greater flexibility to Excel users who frequently compare worksheet data.

To illustrate four reasons why Diff Doc can be a good alternative to Excel's built-in comparison features, we'll:

- Discuss the limitations of having an older version of Excel when it comes to document comparison.
- Illustrate Diff Doc's reporting capabilities so you can analyze the differences between documents.
- Point out other file types that you can compare with an Excel file to cut down on needlessly converting documents.
- Demonstrate how Diff Doc can compare the files in a folder, instead of comparing the individual files themselves.

Application:
Microsoft Excel 2000/2002/2003

Operating Systems:
Microsoft Windows

Keywords:
Excel, Compare, Side By Side, Utility, Diff Doc

Download:
http://download.elementkjournals.com/excel/200512/diffdoc.zip

Most data analysis entails comparing one workbook with another. But when you're comparing several workbooks (or worksheets), it isn't efficient to switch back and forth between them. And, the Compare Side By Side feature is new to Excel; if you use an older version, it may not even be an option. SoftInterface, Inc.'s Diff Doc utility provides a flexible interface for comparing Excel documents that may be a great alternative to Excel 2003's built-in Compare Side By Side feature.

Reason #1: You don't use Excel 2003

In the December 2004 issue of *Inside Microsoft Excel*, our article "Simplify comparison tasks by synchronizing worksheet windows" outlined the new Compare Side By Side feature in Excel 2003. However, if you haven't upgraded to 2003, you don't have that option for flexible document comparison. Diff Doc breaks down the version obstacle by accepting any file with an .xls filename extension.

However, Mac and Unix users are out of luck since Diff Doc only supports the Windows operating system.

Note: Even though Diff Doc is a free utility, you're encouraged to purchase it with a pop-up reminder. Click Done to dismiss it, but be warned that it appears every 15 minutes until you purchase the software.

Reason #2: It has reporting capabilities

Diff Doc's most important feature is its ability to not only compare two workbooks, but also to quickly report and email the differences. Let's illustrate how it all works.

Compare two workbooks

First, you need to download Diff Doc from SoftInterface Inc.'s website. Then, a quick wizard takes you through the installation.

To download and install Diff Doc:

1. Launch your web browser and go to **www.softinterface.com**.
2. Click the Download button in the Diff Doc section of the website.
3. Type your email address in the E-mail text box and click the Continue To Download button.
4. Click the Download button on the next page and then click Run in the File Download – Security Warning dialog box if necessary.
5. Follow the instructions for the Diff Doc Setup Wizard, selecting the program's location and folder name as you desire.

Once you finish the wizard, you can launch Diff Doc from Windows' Start menu (or from the desktop if you chose to add an icon to it during installation). When you first launch Diff Doc,

Home in on Diff Doc's Excel settings for ultimate customization

by Kara Soos

Application: Microsoft Excel 2000/2002/2003
Operating System: Microsoft Windows
Keywords: Excel, Diff Doc, Compare, Settings, Preferences

In addition to the general settings for Diff Doc, there are also settings specific to Excel workbooks. Diff Doc's flexibility is primarily a result of these preferences.

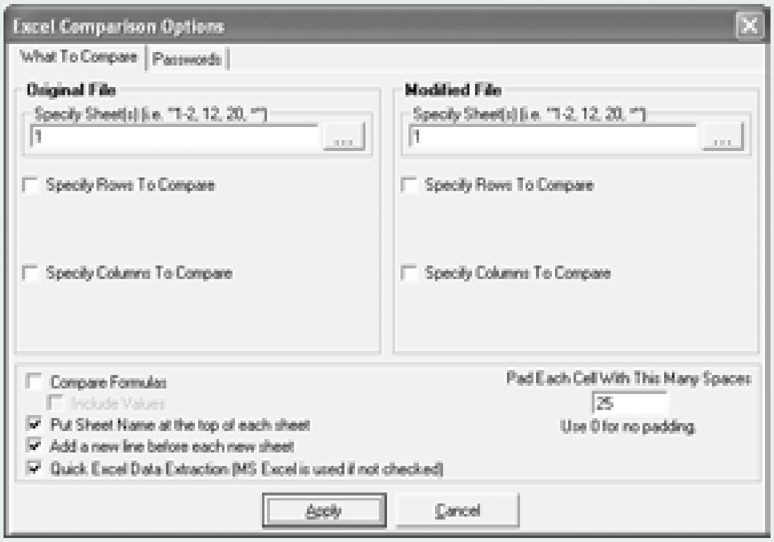
To change the Excel workbook settings:

1. Click on the Compare Options tab in the Diff Doc window.
2. Select the General tab, if necessary.
3. Click the Excel button in the Compare Options Specific To File Type panel.

In the resulting Excel Comparison Options window, as shown in **Figure A**, you get down to the essentials of your document comparison.

Most of the settings in the Excel Comparison Options window are self-explanatory, but it's important to know the possibilities for customization. You can change the following settings:

- Which rows and/or columns you compare in both the Original and Modified workbooks.
- Which worksheet(s) you compare in both the Original and the Modified workbooks.
- Whether you want to compare formulas in addition to static data.
- How much *padding*, or empty space, you want to appear between worksheet columns.
- Whether the worksheet name appears above its data so you can distinguish where one worksheet ends and the next begins.
- Passwords necessary to open the Original and/or Modified workbooks.



Original File
Specify Sheet(s) (i.e. "1-2, 12, 20, *"): 1
□ Specify Rows To Compare
□ Specify Columns To Compare

Modified File
Specify Sheet(s) (i.e. "1-2, 12, 20, *"): 1
□ Specify Rows To Compare
□ Specify Columns To Compare

□ Compare Formulas
□ Include Values
☑ Put Sheet Name at the top of each sheet
☑ Add a new line before each new sheet
☑ Quick Excel Data Extraction (MS Excel is used if not checked)

Pad Each Cell With This Many Spaces: 25
Use 0 for no padding

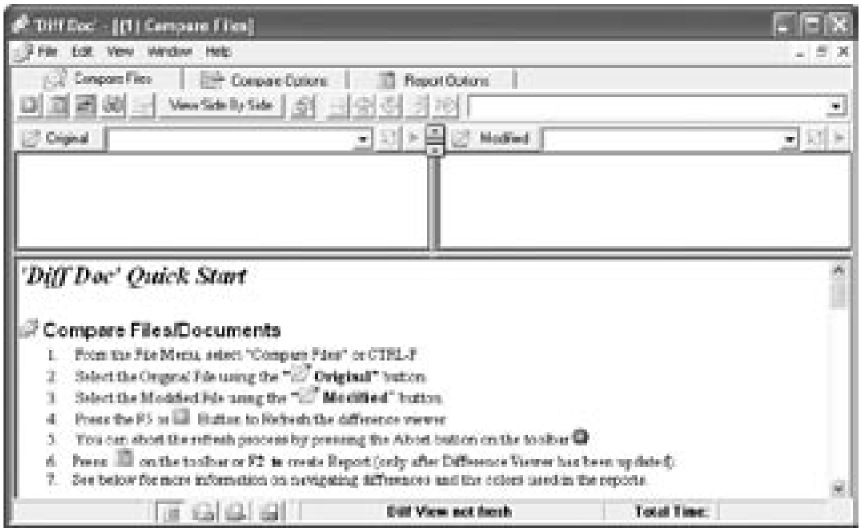
A: You can change the settings for comparing Excel workbooks to eliminate unnecessary information and make the differences easier to read.

the program's interface displays, as shown in **Figure A**.

Note: You can follow along with our example by downloading and extracting the sample files from **diffdoc.zip** found at the URL given at the beginning of this article.

To designate two workbooks for comparison:

1. Click the Original button on the left side of the Diff Doc window, navigate to the sample file **January 2005.xls** in the Load File dialog box, and click Open.

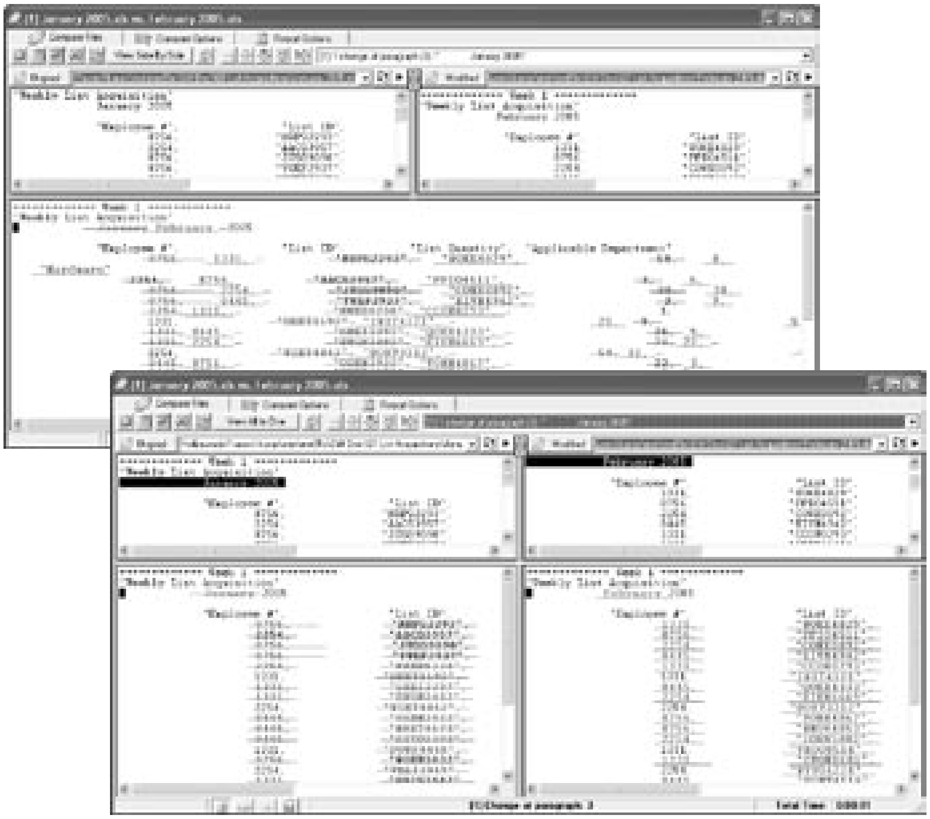


'Diff Doc' Quick Start

Compare Files/Documents

1. From the File Menu, select "Compare Files" or CTRL-F
2. Select the Original File using the " Original" button
3. Select the Modified File using the " Modified" button
4. Press the F5 or Button to Refresh the difference viewer
5. You can abort the refresh process by pressing the Abort button on the toolbar
6. Press on the toolbar or F2 to create Report (only after Difference Viewer has been updated)
7. See below for more information on navigating differences and the colors used in the reports

A: When you first launch Diff Doc, the Quick Start text appears in the lower half of the window to help you get started.



B: You can view the document's differences in one large window, or you can compare them side by side.

2. Click the Modified button on the right side of the Diff Doc window, navigate to the sample file **February 2005.xls** in the Load File dialog box, and click Open.

3. Press [F5] to show the differences between the two workbooks in the lower window, where the Quick Start text originally displays.

4. Click Yes, and then click OK twice, when Diff Doc prompts you to turn off the Word Wrap feature.

Your Diff Doc window should now look similar to the one shown on the top in **Figure B**. If you prefer to view your differences side by side, as shown on the bottom in **Figure B**, just click the View Side By Side button. Currently, Diff Doc's defaults are all in place and the results look confusing. To customize the display to suit your Excel needs, check out the accompanying article, "Home in on Diff Doc's Excel settings for ultimate customization."

Tip: We found that Excel data tends to truncate after a comma in a number. To avoid this, just click on the Compare Options tab and click the Excel button that appears on the General tab. Then, deselect the Quick Excel Data Extraction (MS Excel Is Used If Not Checked) check box and click Apply.

To create a report of your document changes:

1. Click on the Report Options tab in the Diff Doc window.

2. Select the option button that corresponds to the format in which you want the report. We'll select the MS Word option button.

3. Press [F2] to view the report. Microsoft Word launches and the report displays, as shown in **Figure C**.

The format in which you report the differences between your workbooks and/or worksheets depends on how you plan to use the report. Refer to **Table A** to determine which format is best for your situation.

Reason #3: Compare with other file types

Diff Doc supports more than just Excel workbooks. You can also compare files from other applications, such as Word, PowerPoint, and WordPerfect, not to mention PDF and RTF files. No more importing text files from clients outside the company into Excel workbooks just so you can do a side-by-side comparison of the data. You can keep each file type intact to avoid accidental changes.

Remember when you're comparing an Excel workbook to a file from another application that there are application-specific settings. You can customize the comparison by accessing these settings through the Compare Options tab.

Reason #4: Compare folders instead of files

In addition to comparing the data and text in an Excel workbook, you can use Diff Doc to look at the bigger picture. For example, say you put three workbooks—January 2005.xls, February 2005.xls, and March 2005.xls—in the Q1 List Acquisitions folder. A colleague reviews the workbooks in the folder and saves it in her own folder.

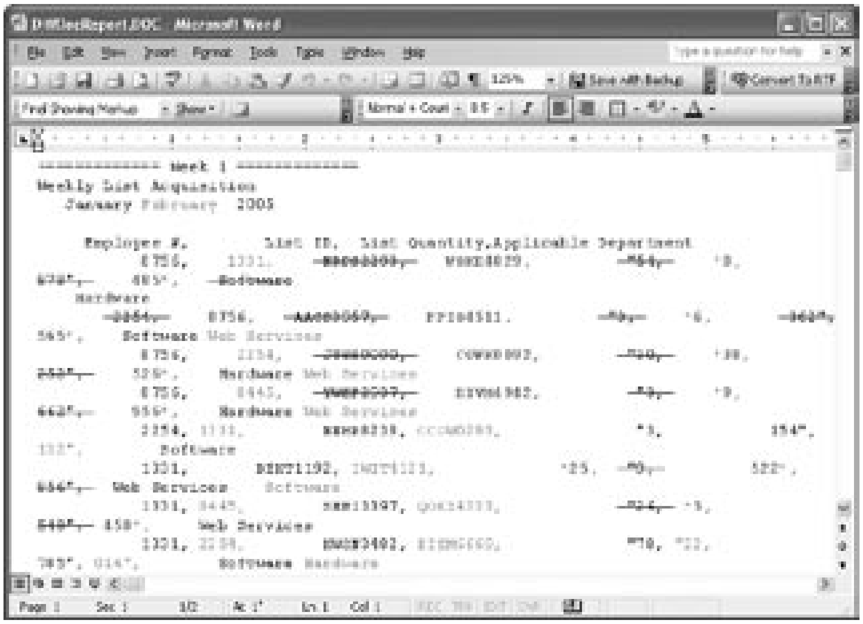
Now you need to see which workbooks she modified. Instead of opening each workbook individually, you can use Diff Doc to compare two folders.

To compare all of the files in two different folders:

1. Click on the Compare Files tab, if necessary.
2. Choose File | Compare Folders from the menu bar to switch to the Compare Files window.
3. Click the Original button on the left side of the window, navigate to the Q1 List Acquisitions folder, and then click OK.
4. Open the modified folder by clicking the Modified button on the right side of the window, navigating to the Copy Of Q1

Table A: Reporting formats and their best uses

Format type	Best to use when ...
MS Word	You want to print results; you need a versatile file format that most users can open; you want to specify viewing only deletions, only additions, or only changes
Text	You aren't concerned with formatting; you want to import and/or export to other programs with ease
Comma Delimited	You want to create a report that can easily be converted to Excel data
HTML	You want to view the report in a web browser; you want to save, print, or email the report from the browser; you want the option of viewing the report all at once or side by side



============= Week 1 =============
Weekly List Acquisition
January February 2005

Employee #, List ID, List Quantity,Applicable Department
8756, 1331, ~~BB000209,~~ WXHE8829, ~~454,~~ *8,
~~678*,~~ 485*, ~~Software~~
Hardware
~~2254,~~ 8756, ~~AA000069,~~ FP108511, ~~40,~~ *6, ~~462*,~~
565*, Software Web Services
8756, 2254, ~~JJ000000,~~ CUWE0892, ~~710,~~ *38,
~~252*,~~ 528*, Hardware Web Services
8756, 8445, ~~VWBF2597,~~ DIVN4982, ~~43,~~ *9,
~~662*,~~ 956*, Hardware Web Services
2254, 1331, BBWB8238, CCGM0281, *5, 154*,
132*, Software
1331, BERT1192, IKIT1121, *25, ~~40,~~ 522*,
~~654*,~~ Web Services Software
1331, 8445, XXM13397, QOK14333, ~~924,~~ *5,
~~540*,~~ 458*, Web Services
1331, 2254, BMGX3402, EIDM6660, *70, *22,
785*, 014*, Software Hardware

C: You can get the results of your workbook comparison as a Word document, an application with which most users are comfortable.

Related article

For more information on Excel's Compare Side By Side feature, check out this article:

- Simplify comparison tasks by synchronizing worksheet windows (December 2004)

This article and more are available to EKJ Online subscribers at **www.elementkjournals.com/premier/journal.asp**.

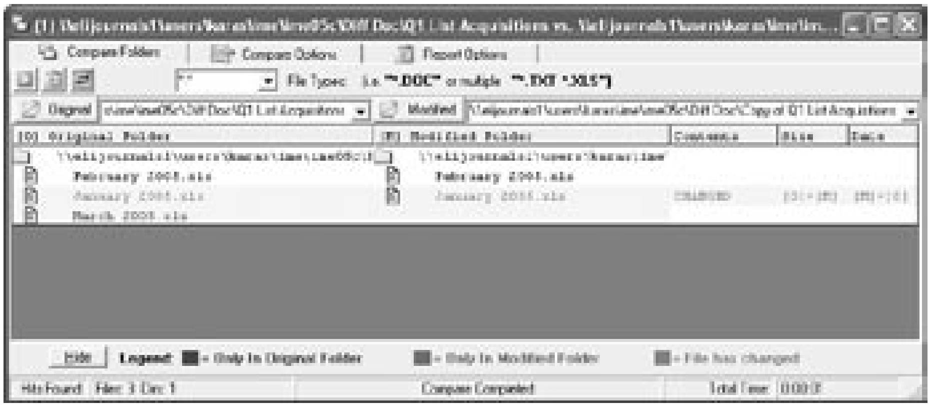
Change the wacky colors and strikethroughs

If the colors and font formatting are too busy for your tastes, you can change how Diff Doc displays additions, deletions, and changes. Just follow these quick steps:

1. Select the Compare Options tab in the Diff Doc window.
2. Click on the View tab and modify the settings in the Additions, Deletions, and Changes panels.
3. Select the Compare Files tab to return to the Diff Doc window and press [F5] to refresh the window with your new settings.

List Acquisitions folder, and then clicking OK.

5. Press [F5] to compare the two folders.



```
February 2005.xls        February 2005.xls
January 2005.xls         January 2005.xls     CHANGED
March 2005.xls
```

D: Diff Doc provides a legend at the bottom of the window so you know exactly what the colors indicate.

As shown in **Figure D**, Diff Doc uses color coding to show you which documents in the folder have changed, which documents only appear in the original folder, and which ones only appear in the modified folder. In addition, if a file has changed, you can see which file (the original or the modified) is larger in size and which is more recent. ✳

The Excelerated Program

Automatically convert currency values into words—and save a ton of data entry time

by Kara Soos

AT A GLANCE

Business checks are no different than personal checks in the sense that you still need to display the amount in two forms: numerical and text. Instead of typing out the English text that corresponds with your check amounts, you can create a few user-defined functions to do it for you.

To save time converting a currency amount into its English text equivalent, we'll:

- Review user-defined functions and discuss how they differ from the macros you know and love.
- Enter the VBA (Visual Basic for Applications) code necessary to create four user-defined functions.
- Step through the important, behind-the-scenes actions that help the functions work together.
- Test the main function and discuss possibilities for customizing it.

Application:
Microsoft Excel 2000/2002/2003/2004

Operating Systems:
Microsoft Windows, Macintosh

Keywords:
Excel, VBA, Macros, Accounting, Select Case, Currency

Download:
http://download.elementkjournals.com
/excel/200512/currencywords.zip

Sometimes the numerical value of your currency data isn't enough; you also need to store the value as English text. For instance, if you're preparing data for checks, you need to display the check's amount in both numerical value and its text equivalent. Forget typing all of that data manually. You can use VBA to create your own user-defined functions, which work together to convert the numbers into words automatically.

Tell me in plain English

User-defined functions are a crucial part of making Excel work for your specific needs. You can transform any calculation that you find yourself performing repetitively into an Excel function, as long as you know how to express it in VBA.

In our example, we need to show the client balances in our spreadsheet in both numerical value and English text, as shown in **Figure A**. This makes it easier when we import the data to an Access database or check-generating software.

Note: You can follow along with our example by downloading and extracting the sample files from **currencywords.zip**, found at the URL given at the beginning of this article.

Define your own functions

Familiar functions, such as the `SUM` or the `IF` functions, are part of Excel. You don't necessarily need to know how they work. However, if you're creating your own functions, you need to understand them on a deeper level. We'll take you through the four functions described in **Table A**, which work together to convert numbers into words.

User-defined functions differ from the procedures you're used to seeing. Macros often use procedures (the familiar `Sub…End Sub` structure), but a function works a little differently. A function doesn't perform a series of tasks, but it takes a value or a cell reference and passes it through a calculation. When you create a user-defined function, you're essentially recording the steps of the calculation. Then, you can use the function just as you would any other built-in Excel function. You'll see how it all comes together as we step through this article.

Start with the main SpellCurrency() function

Instead of boring you with a play-by-play commentary on each function, we'll summarize each one and highlight the important points you may want to note.

To create the main SpellCurrency() function:

1. Launch Excel and open the file in which you want to use the function. If you're following our example, open the sample file **checkdata.xls**.

2. Choose Tools | Macro | Visual Basic Editor to open the VBE (Visual Basic Editor). Or, in Windows, you can press [Alt][F11].

3. Select Insert | Module from the VBE's menu bar to open a new module.

4. Enter the code shown in **Listing A**.

Note: The code for each of the listings in this article is part of the download file. Just open the sample file **vba.txt** and then copy and paste it into the VBE. This should save you from re-entering the data and possibly making typos.

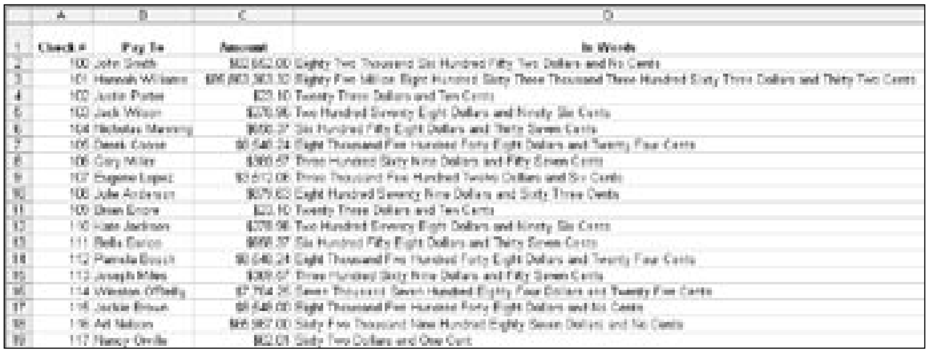
Look closer at the SpellCurrency() function

The `SpellCurrency()` function brings it all together. It's the skeleton for the other three functions, as they focus on more specific tasks. This is the function you'll need to type into a cell to actually perform the function, whereas the other three functions work behind the scenes. Mostly, the function uses String functions (`Left`, `Right`, `Mid`, `Len`, and `Trim`) to find the decimal place in the value and assign it text based on its position. We'll cover String functions in more detail later in the article. Here are two other aspects of the `SpellCurrency()` function that are important to note:

- The variables defined in the beginning of the function include a `Redim` statement. You use `Redim` statements to define a flexible array. In our example, the `Redim` statement helps us determine whether the value is in the trillions, billions, millions, or thousands depending on the placement of the decimal point. You can also add `Preserve` to a `Redim` statement, which means that only the last dimension of the array can change. Since our array only has one dimension, it isn't necessary.
- The `ByVal` expression designates parameters that pass a value so that the variable takes on this value temporarily without permanently altering the original variable. It's as if a copy of the variable passes through the procedure as a value, but it doesn't affect the variable itself. In this case, the `ByVal` expression is just temporary information that passes through the function without actually altering it.

Convert the hundreds place

The `ConvertHundreds()` function works by utilizing String functions along with an `If…Then…Else` statement. The same `ByVal` value is passed through the function, which is the temporary text string representing the currency value.



	A	B	C	D
1	Check #	Pay To	Amount	In Words
2	100	John Smith	$82,652.00	Eighty Two Thousand Six Hundred Fifty Two Dollars and No Cents
3	101	Hannah Williams	$85,863,363.32	Eighty Five Million Eight Hundred Sixty Three Thousand Three Hundred Sixty Three Dollars and Thirty Two Cents
4	102	Justin Porter	$23.10	Twenty Three Dollars and Ten Cents
5	103	Jack Wilson	$278.96	Two Hundred Seventy Eight Dollars and Ninety Six Cents
6	104	Nicholas Manning	$658.37	Six Hundred Fifty Eight Dollars and Thirty Seven Cents
7	105	Derek Coone	$8,548.24	Eight Thousand Five Hundred Forty Eight Dollars and Twenty Four Cents
8	106	Cory Miller	$369.57	Three Hundred Sixty Nine Dollars and Fifty Seven Cents
9	107	Eugene Lopez	$3,512.06	Three Thousand Five Hundred Twelve Dollars and Six Cents
10	108	Julie Anderson	$879.63	Eight Hundred Seventy Nine Dollars and Sixty Three Cents
11	109	Dean Emore	$23.10	Twenty Three Dollars and Ten Cents
12	110	Kate Jackson	$278.96	Two Hundred Seventy Eight Dollars and Ninety Six Cents
13	111	Bella Carson	$658.37	Six Hundred Fifty Eight Dollars and Thirty Seven Cents
14	112	Pamela Bosch	$8,548.24	Eight Thousand Five Hundred Forty Eight Dollars and Twenty Four Cents
15	113	Joseph Miles	$369.57	Three Hundred Sixty Nine Dollars and Fifty Seven Cents
16	114	Winston O'Reilly	$7,784.25	Seven Thousand Seven Hundred Eighty Four Dollars and Twenty Five Cents
17	115	Jackie Brown	$8,548.00	Eight Thousand Five Hundred Forty Eight Dollars and No Cents
18	116	Art Nelson	$65,987.00	Sixty Five Thousand Nine Hundred Eighty Seven Dollars and No Cents
19	117	Nancy Orville	$62.01	Sixty Two Dollars and One Cent

A: You don't need to manually type this cumbersome text when you define functions to create it automatically.

Table A: Four user-defined functions and their tasks

Function name	Listing	Purpose
`SpellCurrency()`	A	Main function; brings together the other three functions; adds the "dollars" and "cents" text.
`ConvertHundreds()`	B	Finds the hundreds place, if applicable, and adds the "hundred" text to it; works with the `ConvertTens()` and `ConvertDigit()` functions.
`ConvertTens()`	C	Replaces the tens place with text to represent either the numbers 1-19 or 10-99.
`ConvertDigit()`	D	Replaces single digits with the text for numbers 1-9, if applicable.

You can include the code for the ConvertHundreds() function by leaving a blank line in the VBE's code window, and then entering the code in **Listing B**. Or, if you're using the TXT file we provide, you can simply copy and paste it into the code window.

There are two If...Then...Else statements in the ConvertHundreds() function. The first one has only one condition, and therefore it omits the optional Else argument. If the digit in the hundreds place isn't zero, then the ConvertDigit() function is called to replace the digit with its text equivalent, and then add the text *hundred* to it.

The second If...Then...Else statement has two conditions. If the digit to the right of the hundreds place (the tens place) isn't zero, then the ConvertTens() function is called to generate the

Listing A: The SpellCurrency() function

```
Option Explicit
Function SpellCurrency(ByVal MyText)
Dim Dollars, Cents, Temp
Dim DecimalPlace, Count
ReDim Place(9) As String

Place(2) = " Thousand "
Place(3) = " Million "
Place(4) = " Billion "
Place(5) = " Trillion "
MyText = Trim(Str(MyText))

DecimalPlace = InStr(MyText, ".")
    If DecimalPlace > 0 Then
        Cents = ConvertTens(Left(Mid(MyText, _
        DecimalPlace + 1) & "00", 2))
        MyText = Trim(Left(MyText, DecimalPlace - 1))
    End If
    Count = 1

    Do While MyText <> ""
        Temp = ConvertHundreds(Right(MyText, 3))
        If Temp <> "" Then
            Dollars = Temp & Place(Count) & Dollars
        If Len(MyText) > 3 Then
            MyText = Left(MyText, Len(MyText) - 3)
        Else
            MyText = ""
        End If
        Count = Count + 1
    Loop

    Select Case Dollars
        Case "": Dollars = "No Dollars"
        Case "One": Dollars = "One Dollar"
        Case Else: Dollars = Dollars & " Dollars"
    End Select

    Select Case Cents
        Case "": Cents = " and No Cents"
        Case "One": Cents = " and One Cent"
        Case Else: Cents = " and " & Cents & " Cents"
    End Select
    SpellCurrency = Dollars & Cents
End Function
```

Related articles

- Avoid major business errors: Double-check totals with a UDF (February 2002)
- Simplify worksheet formulas with user-defined functions (June 2001)

These articles and more are available to EKJ Online subscribers at **www.elementkjournals.com/premier/journal.asp**.

Listing B: The ConvertHundreds() function

```
Function ConvertHundreds(ByVal MyText)
Dim Result As String

    If Val(MyText) = 0 Then Exit Function
    MyText = Right("000" & MyText, 3)

    If Mid(MyText, 1, 1) <> "0" Then
        Result = ConvertDigit(Mid _
            (MyText, 1, 1)) & " Hundred "
    End If

    If Mid(MyText, 2, 1) <> "0" Then
        Result = Result & ConvertTens(Mid(MyText, 2))
    Else
        Result = Result & ConvertDigit(Mid(MyText, 3))
    End If
    ConvertHundreds = Result
End Function
```

appropriate text. Otherwise, the ConvertDigit() function is called.

As these functions begin to come together, you can see how they intertwine and feed off one another to work toward the goal of translating digits into words. A vital part of this task is accomplished through String functions, which we'll go over next.

Identify decimal placement with String functions

String functions are really the backbone of these four functions. By using the Left, Right, Mid, Len, and Trim functions, our four functions can identify where digits are in relation to the decimal point, and therefore assign the correct text to reflect the numerical value. For a quick review of these String functions and how they work, check out **Table B**.

Use Select Case to substitute words

The first two functions rely on these final functions: ConvertTens() and ConvertDigit(). You can enter the code for these functions in the VBE code window by referring to **Listings C** and **D**. Or, you can copy and paste the code from our sample TXT file.

These two functions are composed of mainly a Select Case statement. These statements set up several conditions, each one with a corresponding result. In our case, each numerical value is assigned a text string. The ConvertDigit() function is the simplest example, since it merely converts the single digits (1-9) into their English text equivalent. The ConvertTens() function is a bit more complicated. It uses Select Case in addition to an If...Then...Else statement. It converts the numbers 10 through 19 into text if the value of the tens place is one. Otherwise, it converts the values 2-9 into their tens place equivalents, 20 through 90.

Select Case vs. If...Then... Else statements

You've probably noticed that these two statements accomplish very similar tasks—and it's true. Both If...Then...Else statements and Select Case statements set up conditions and what result to return based on those conditions. However, Select Case is more efficient when:

- the same variable or expression appears on the left side

Table B: String functions and their uses

Function	Syntax	Use it when you need to ...
Left	Left(string, number)	Find the characters to the left of a particular placement in the text string
Right	Right(string, number)	Find the characters to the right of a particular placement in the text string
Mid	Mid(string, start, length)	Find the characters in the middle of a text string, starting from a particular placement and ending after a designated number of characters
Len	Len(string)	Return the number of characters in a text string (or cell reference)
Trim	Trim(string)	Eliminate any extra leading spaces from a text string (or cell reference)

Listing C: The ConvertTens() function

```
Function ConvertTens(TensText)
Dim Result As String
Result = ""

    If Val(Left(TensText, 1)) = 1 Then
        Select Case Val(TensText)
            Case 10: Result = "Ten"
            Case 11: Result = "Eleven"
            Case 12: Result = "Twelve"
            Case 13: Result = "Thirteen"
            Case 14: Result = "Fourteen"
            Case 15: Result = "Fifteen"
            Case 16: Result = "Sixteen"
            Case 17: Result = Seventeen"
            Case 18: Result = "Eighteen"
            Case 19: Result = "Nineteen"
            Case Else
        End Select
    Else
        Select Case Val(Left(TensText, 1))
            Case 2: Result = "Twenty "
            Case 3: Result = "Thirty "
            Case 4: Result = "Forty "
            Case 5: Result = "Fifty "
            Case 6: Result = "Sixty "
            Case 7: Result = "Seventy "
            Case 8: Result = "Eighty "
            Case 9: Result = "Ninety "
            Case Else
        End Select
        Result = Result & ConvertDigit _
            (Right(TensText, 1))
    End If
    ConvertTens = Result
End Function
```

- the conditions aren't too complicated (no `And` or `If` operators) and
- they don't necessitate a specific order.

In general, when you have more than three conditions to define, an `If…Then…Else` statement gets cumbersome and difficult to read.

To test the SpellCurrency() function:

1. Close the VBE and return to your workbook.
2. Select cell D2 and enter `=SpellCurrency(C2)`.
3. Press [Enter] to view the results.

Now you can use this user-defined function as you would any other. Copy it to adjacent cells by dragging the Fill handle. Or, instead of a cell reference, enter a numerical value, such as `=SpellCurrency("$523.21")`.

Listing D: The ConvertDigit() function

```
Function ConvertDigit(Digit)
      Select Case Val(Digit)
            Case 1: ConvertDigit = "One"
            Case 2: ConvertDigit = "Two"
            Case 3: ConvertDigit = "Three"
            Case 4: ConvertDigit = "Four"
            Case 5: ConvertDigit = "Five"
            Case 6: ConvertDigit = "Six"
            Case 7: ConvertDigit = "Seven"
            Case 8: ConvertDigit = "Eight"
            Case 9: ConvertDigit = "Nine"
            Case Else: ConvertDigit = ""
      End Select
End Function
```

Tweak the code to match your needs

Although these functions don't leave much room for adjustment, you can always modify the text to suit different types of currency. For example, if your company deals with British clients and you send checks in their native currency, you can replace all "dollar" and "dollars" text in the code with "pound" and "pounds," and then replace the "cent" and "cents" text with "pence." Then, the translated text can match the symbols that may result from working with foreign currency. If you do make these adjustments, you may want to change the name of the function to reflect which type of currency it translates. In theory, you could replicate these four functions in different currencies, and then you'd have user-defined functions to reflect each one.

Another way you can use these functions is to translate the currency into a different language, if you so desire. All you'd need to do is replace the English text with the new language, such as Spanish or French. Just be careful of different language conventions that may change the values' translation. ✳

⌛ Quick Tips

Let Excel format your PivotTable so you can concentrate on the data (2000/2002/2003/2004)

Keywords: Excel, PivotTable, AutoFormat, Tips

PivotTables have a job to do; they don't always have to look pretty. As long as they're helping you get to the bottom of your data, that's all you need. However, there are situations when you want your PivotTable to look presentable and appealing, such as when you need to show your supervisor the latest budget figures.

Instead of spending valuable time trying to pretty-up your PivotTable, let Excel do the work for you.

To AutoFormat your PivotTable:

1. Launch Excel and open the file in which you have a PivotTable that you'd like to format.
2. Select the worksheet that contains your PivotTable and click anywhere inside the table to activate it.
3. Choose View | Toolbars | PivotTable to display the PivotTable toolbar, if necessary.
4. Click the Format Report button on the PivotTable toolbar to open the AutoFormat window.
5. Select a format and click OK to apply it.

Note that you can choose to format the PivotTable in the form of a report or a table.

Authenticate your macros so colleagues know the workbook is safe to open

by T. L. Thomas and Kara Soos

AT A GLANCE

Excel workbooks with macros are often an overlooked entry for viruses, especially when you don't know who created the macro. You can verify that macros come from trusted sources by assigning a digital signature to your macro projects.

To enforce a proactive security method by digitally signing our macros, we'll:

- Specify a security level to filter safe and unsafe macros.
- Obtain and install a digital certificate from a commercial certification authority.
- Create your own digital certificate for personal use.
- Add a digital signature to a macro project.

Applications:
Microsoft Excel 2000/2002/2003, Microsoft Internet Explorer 5.0 or later

Operating System:
Microsoft Windows

Keywords:
Excel, Macros, Security, Sign, Certificates, Signatures

Hackers continually challenge themselves to come up with new ways to break through company firewalls and wreak havoc on innocent users' systems. Many companies employ developers dedicated to creating macro solutions for company-wide use. However, the more widely distributed a macro solution is and the bigger the company is that employs it, the lower the likelihood is that you know who created the macro. You can protect your macros and the people who use them by signing them with a digital signature based on a digital certificate, as shown in **Figure A**.

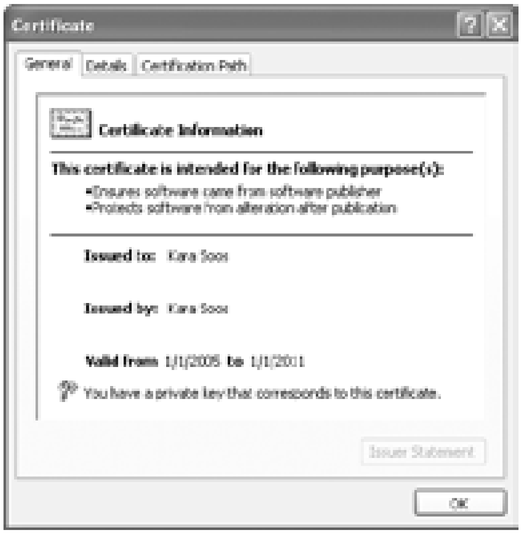
Set your macro security level

There are three macro virus protection security levels in Excel.

- **Very High (2003 only).** The Very High security setting only allows macros installed in a trusted location, whether they're signed or unsigned. You must manually include the source in a list of trusted publishers, which we'll show you later in the article.
- **High.** The High security setting protects against any macro that isn't digitally signed by automatically disabling any unsigned macros.
- **Medium.** The Medium security setting prompts you to choose whether to run macros by presenting the Security Warning dialog box.
- **Low.** The Low security setting turns off macro virus protection altogether.

To set your macros security level:

1. Launch Excel and open the workbook that contains macros you wish to certify.
2. Choose Tools | Macro | Security from the menu bar to access the Security window, as shown in **Figure B**.



Certificate Information

This certificate is intended for the following purpose(s):
•Ensures software came from software publisher
•Protects software from alteration after publication

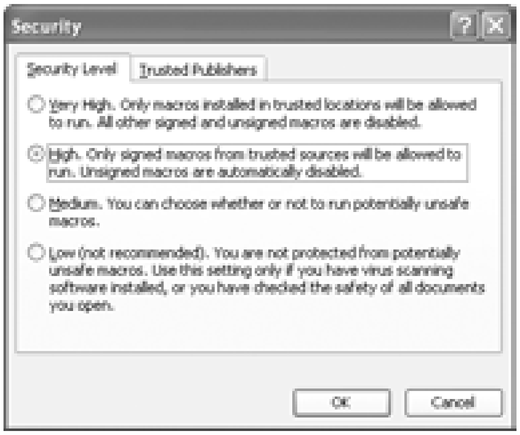
Issued to: Kara Soos

Issued by: Kara Soos

Valid from 1/1/2005 to 1/1/2011

You have a private key that corresponds to this certificate.

A: You can verify that the macros in your distributed workbook are safe by assigning them your digital certificate.



○ Very High. Only macros installed in trusted locations will be allowed to run. All other signed and unsigned macros are disabled.

● High. Only signed macros from trusted sources will be allowed to run. Unsigned macros are automatically disabled.

○ Medium. You can choose whether or not to run potentially unsafe macros.

○ Low (not recommended). You are not protected from potentially unsafe macros. Use this setting only if you have virus scanning software installed, or you have checked the safety of all documents you open.

B: Microsoft recommends that you set your macros security setting to High, which disables all unsigned macros.

3. Select the Security Level tab, if necessary, and then select the option button corresponding to the security level you desire.

4. Click OK to apply the new security setting.

Authenticode technology

You've probably seen the familiar dialog box that appears when you open a workbook with macros. You may click whatever it takes to dismiss the window, without stopping to think about what it means for a workbook to contain unsigned macros.

Microsoft Office's enhanced macro security protection utilizes Authenticode technology to verify a macro's source before you run it. Authenticode technology refers to the macro's digital certificate; it makes sure that the program code was signed by a verified source and wasn't tampered with since its signing. Authenticode technology is also used on the internet to confirm the safety of downloadable software.

Commercial certification vs. individual certification

You can request a digital certificate, also known as an SPC (Software Publishing Certificate) for commercial or individual purposes. For both kinds of requests, you need to supply personal information so the Certification Authority can verify your identity and credentials. You also need to take a pledge stating that you won't knowingly distribute software infected with viruses. The difference between the two certifications is that commercial requests also need to meet a specific Dun & Bradstreet rating, which verifies the corporation's financial standing and stability. This merely helps the CA clearly distinguish between individual and commercial certificates.

Obtain a digital certificate

There are two ways to obtain a digital certificate: create one yourself or request one from a Certification Authority (CA). When you assign a self-signed certificate to a macro project, users continue to receive a security warning that prompts them to enable or disable macros until they specify that you're a reliable source. However, when you assign an authenticated certificate to a macro project, the CA—a reliable source in its own right—verifies you as a reliable source to users.

Request an authenticated certificate

You can request an authenticated digital certificate from a Microsoft-approved CA. That way, you can sign any of your macro projects with it to reassure users that it's genuine and safe. To obtain a list of Microsoft-approved CAs, check out **http://msdn.microsoft.com/library/?url=/library/en-us/dnsecure/html/rootcertprog.asp**.

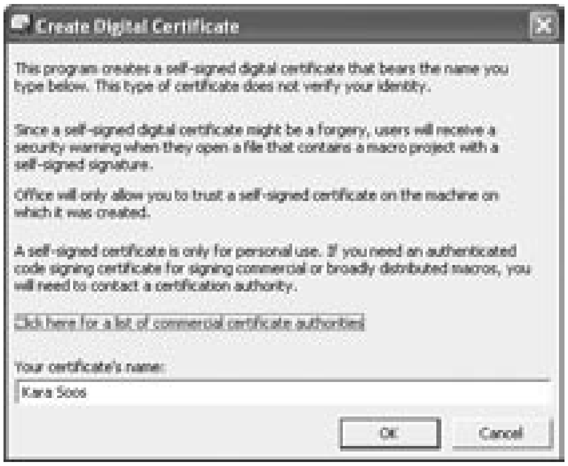
Create your own digital certificate

You can create a digital certificate with Microsoft Office's SelfCert tool. However, the SelfCert tool creates a self-signed digital certificate that doesn't verify your identity. You should only employ it for personal use because, hypothetically, anyone can unsign a macro project and add his own digital signature unless you request a digital certificate from a CA.

To create a digital signature in your name:

1. Search for the selfcert.exe file on your hard drive, using your operating system's Search feature and double-click on the file to run it. If you don't have this file anywhere on your hard drive, see the "Install the SelfCert tool" shaded box.

2. Type your name in the Your Name text box when the Create Digital Certificate dialog box appears, as shown in **Figure C**.


Create Digital Certificate
Click here for a list of commercial certificate authorities
OK
Cancel

This program creates a self-signed digital certificate that bears the name you type below. This type of certificate does not verify your identity.

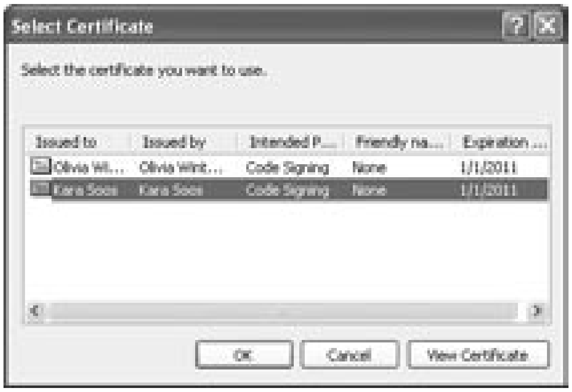
Since a self-signed digital certificate might be a forgery, users will receive a security warning when they open a file that contains a macro project with a self-signed signature.

Office will only allow you to trust a self-signed certificate on the machine on which it was created.

A self-signed certificate is only for personal use. If you need an authenticated code signing certificate for signing commercial or broadly distributed macros, you will need to contact a certification authority.

Your certificate's name: Kara Soos

C: You can create your own digital certificate, but it doesn't verify your identity and credentials the way a Certification Authority does.



Select the certificate you want to use.

Issued to	Issued by	Intended P...	Friendly na...	Expiration ...
Olivia Wi...	Olivia Wint...	Code Signing	None	1/1/2011
Kara Soos	Kara Soos	Code Signing	None	1/1/2011

D: You can view further details about a certificate by selecting it and clicking the View Certificate button.

3. Click OK to create the digital certificate, and then click OK again when Windows verifies that the operation is successful.

Assign your digital signature to a macro

Now that you've created a digital certificate, whether self-created or requested, you can use it to make your mark on any macro project.

To digitally sign a macro and view the new certificate:

1. Switch back to your Excel workbook.
2. Press [Alt][F11] to access the VBE (Visual Basic Editor) or you can choose Tools | Macro | Visual Basic Editor from the menu bar.
3. Select the project you want to sign in the Project Explorer.
4. Choose Tools | Digital Signature from the VBE's menu bar to open the Digital Signature dialog box.
5. Click Choose to access the Select Certificate dialog box, and then select your certificate from the list box, as shown in **Figure D**.
6. Take a look at the certificate by clicking the View Certificate button.
7. Dismiss any open dialog boxes and return to your Excel workbook.

Make sure your certificate is among trusted publishers

If you view the certificate for your newly assigned digital signature, it may indicate that you need to store the certificate in your Trusted Root Certification Authority Store. And you're probably thinking, "What is that all about?" You can access the publishers that your system automatically trusts, and then include your personal certificate among them to complete the certificate, as shown in **Figure A**.

Install the SelfCert tool

If the selfcert.exe file isn't present, you may need to install it. To do so, follow these steps:

1. Access Windows Control Panel and double-click on Add Or Remove Programs (or Add/Remove Programs depending on your operating system).
2. Select Microsoft Office (or Microsoft Excel if you use it as a standalone program) from the Currently Installed Programs list box.
3. Click the Change button to run the Microsoft Office setup.
4. Choose Add Or Remove Features option button in the Maintenance Mode Options window and click Next.
5. Select the Choose Advanced Customization Of Applications check box and click Next.
6. Drill down to the Microsoft Office | Office Shared Features directory and click on the Digital Signature For VBA Projects icon. (In some operating systems, you may need to look in the Microsoft Office | Office Tools directory.)
7. Select Run From My Computer from the resulting shortcut menu, and then click the Update button.

You may be prompted to insert your Microsoft Office or Microsoft Excel installation disc.

To export your personal certificate to the Desktop:

1. Launch Internet Explorer and then choose Tools | Internet Options from the menu bar.
2. Select the Content tab and click the Certificates button in the Certificates panel.
3. Choose the Personal tab, if necessary, and select your certificate from the list box.

Inside Microsoft Excel

Inside Microsoft Excel (ISSN 1075-1580) is published monthly by Eli Journals, Inc., a subsidiary of Eli Research, 2222 Sedwick Road, Durham, NC 27713.

Customer Relations & Subscription Services

New Hill Services
Dept. 1380
Denver, CO 80291-1380
Email: service@medville.com

U.S. toll-free	(800) 508-2582
Outside U.S.	(239)-280-2383
Fax	(800) 508-2592

Editorial

Editor	Kara Soos Email: ime_editor@elementk.com Toll-free: (866) 614-8715
Copy Editor	Teena Artman
Contributing Editors	Frank Kuhn T. L. Thomas
Graphic Designers	Rachel Allen Pam Nichols
Creative Director	Chuck Willey
Managing Editor	Craig Watkins
Editorial Director	Erin Lang Bonin, PhD

Publishing & Sales

Publisher	Sherry Kappel (888) 779-3718, ext. 348
Bulk Sales.	Pam Snyder (877) 853-9938
Customer Service Manager	Holly Smith

Postmaster

Postmaster: Send address changes to

Inside Microsoft Excel
P.O. Box 92880
Rochester, NY 14692

Price

Domestic	$127/yr.
Outside the U.S.	$137/yr.

GST #: 889037958. PM #: 40031405.

Back Issues

To order a back issue, call Customer Service at (800) 874-9180. You can pay with MasterCard, VISA, Discover, or American Express.

Printed in the U.S.A.

Are you moving?

If you've moved recently or you're planning to move, you can guarantee uninterrupted service on your subscription by calling us at (800) 874-9180 and giving us your new address. Or you can fax us your label with the appropriate changes at (585) 789-3560. Our Customer Relations department is also available at http://www.eliresearch.com/contact.htm.

To see a list of our products, visit our websites at www.elementkjournals.com and www.eliresearch.com.

Coming up…

- Use a pop-up calendar to control date entry
- Streamline worksheet formulas with arrays

4. Click the Export button to launch the Certificate Export Wizard (Certificate Manager Export Wizard depending on your operating system) and click Next to advance through the wizard.
5. Ensure that the No, Do Not Export The Private Key option button is selected as you go through the wizard.
6. Type a meaningful name for your certificate in the File To Export screen's text box, and then click Next and Finish.
7. Click OK to dismiss the confirmation of a successful export.

Once you export the certificate, it appears on your Desktop. Now, you need to import it to your Trusted Root Certification Authorities Store.

To import the certificate to designate it as trusted:

1. Click the Import button in the Certificates dialog box to access the Certificate Import Wizard. (The same certificate should still be selected.)
2. Read through the wizard's introduction and click Next.
3. Click the Browse button and navigate to the certificate file you just saved on the Desktop, select it, and click Open.
4. Advance in the wizard by clicking Next and ensure that the Place All Certificates In The Following Store option button is selected.
5. Click Browse, select the Trusted Root Certification Authorities folder, and click OK to dismiss the Select Certificate Store dialog box.
6. Click Next and Finish to complete the import. Click Yes or OK if you're prompted to confirm the import.

When you view the certificate through the Select Certificate dialog box, it should now list the certificate's intentions, as shown in **Figure A**. Also, any workbook containing macros signed with your signature won't trigger a security warning when you open it.

Know the limitations of a digital signature

Digital signatures, particularly ones assigned by CAs, are important for anyone who wants to distribute macros in their Excel workbooks. However, there are important limitations that you should understand.

- Whenever you modify the code in a signed macro project, the digital signature is removed. However, as long as you've installed the source's digital certificate on your system, the macro is re-signed each time you save it. Be sure to test your macro before signing it.
- To prevent users from modifying your code and, as a result, invalidating your signature, it's a good idea to lock your macro before you sign it. This way, users can still modify a workbook that implements the macro, but they can't modify the macro itself.
- Other developers can replace your signature with their own at any time. A digital signature is no more than an honorable guarantee that you approved the code. However, a signature with a certificate that's approved by a CA carries more weight and validity. ✳

Inside Microsoft Excel 2005 index

Can't find the article you need? Look no further. The complete 2005 index for *Inside Microsoft Excel* is now available online! Use the searchable online index to find any article, any time at **www.elementkjournals.com/index_ime.asp**. The online index is arranged by month and includes a list of article titles for all issues published in *Inside Microsoft Excel* during 2005. The 2005 index is a great problem-solving resource when you need answers—fast! Plus, you can locate articles of interest in issues missing from your collection. To order back issues, call Customer Relations at (800) 874-9180. Or, to gain online access to articles prior to 2005, please visit **www.elementkjournals.com/premier/premierInfo.asp** for more information.