

50 essential Excel tips

Your users can save a lot of time and effort working with Excel if they know a few tricks and shortcuts. These valuable tips originally appeared in our Excel e-newsletter, which has now been beefed up to include productivity tips for the entire Microsoft Office suite. [To begin receiving this free, valuable resource every Wednesday, visit our e-newsletter subscription center.](#)

Table of contents

Easy data-posting to the Web.....	3
Open Web pages right in Excel	3
Detect And Repair.....	3
User-level prevention is a magic cure	4
Opening multiple windows	4
Creating custom lists.....	4
Copying styles between workbooks	5
A macro that pulls source data from charts	5
Update for Excel 2000 text export.....	5
Creating a workspace in Excel.....	5
Printing formulas	6
Shifting a scatter chart's axis	6
Recovering data with the SYLK format	6
Faster access to worksheets	7
Inserting the full file path in headers and footers	7
Exceeding the 30-argument limit	7
Ask Mr. Excel	7
Catching data entry errors	7
Quickly setting a print area	8
Counting the days via cell subtraction	8
Quickly shuffling toolbar buttons	8
Opening a workbook at startup	8
Hiding duplicate records	8
Converting Julian dates	9
Saving time with Custom Views.....	9
Beware of sharing dates between Excel for Windows and Excel for Macs	9

Don't use AutoFormat on an entire worksheet.....	10
Keep track of changes	10
Conditional formatting	10
Data entry solutions with the Template Wizard.....	11
Customizing "division by 0" error messages.....	11
Debugging your code.....	11
Debugging using watches.....	12
Ensure proper data entry with validation.....	12
Hyperlink to a specific cell.....	12
Macro security settings	13
Express yourself with comments	13
Map your data—on real maps.....	13
Create an Outlook message from Excel	13
Repairing broken file type associations	14
Web Publishing Wizard Error with Office 2000	14
Use auditing to troubleshoot	15
Using Goal Seek.....	15
Using shared workbooks.....	15
Use the Immediate window.....	15
Summarize data with grouping	16
Take a few shortcuts.....	16
Analyzing external data in Excel.....	17
Take advantage of the Personal Macro Workbook.....	17
Clear out all comments	17

Easy data-posting to the Web

If your organization is using an intranet to share public data—and whose organization isn't these days—your users will want an easy way to post this information. Excel 97 offers an easy, wizard-driven system to walk even the most apprehensive users through this process:

1. Select the cell range that contains the data to be published.
2. Select Save As HTML from the File menu.
3. Answer the questions posed by the Internet Assistant Wizard.

The wizard walks users through four steps that define how information will be converted for Web display. Various options include:

- Inserting the resulting HTML tables into an existing HTML page or creating an entirely new page.
- Specifying formatting options, such as rules between data.
- Saving the new HTML page as a freestanding file or inserting it directly into a FrontPage Web.

Open Web pages right in Excel

If your users need to analyze data found on a Web site, they might be in for a bunch of tedious retyping. Under the right circumstances, however, Excel 97 can save them a lot of time by letting them open an HTML page in Excel directly from the Web.

The procedure is just a slight modification of the standard file-opening procedure:

1. Choose Open from the File menu.
2. In the File Name text box, type the URL of the Web file to open. Note: Be sure to include the prefix `http://`.
3. From the Files Of Type drop-down list, select HTML Documents (*.html, *.htm).
4. Click Open.

Excel opens HTML pages based on the HTML table structure of the page, so some Web design treatments may lead to some weird results in Excel. For pages that cleanly present data in a table format, however, this approach can be a real time-saver.

Detect And Repair

Microsoft Office 2000 programs, including Excel, contain a command line called Detect And Repair, which checks and fixes problems with Office program files, DLLs, and registry settings. (This utility does not affect worksheets or other document files.)

Detect And Repair, which you access as a command line option of the Setup utility, performs the following actions:

- Reinstalls Office .exe and .dll files if missing, out-of-date, or corrupt
- Reinstalls all Windows Installer shortcuts, overwriting any existing shortcuts
- Rewrites all required local machine registry values
- Rewrites all required user registry values

You can run Detect And Repair from within Office, but that won't do much good if a serious error is preventing your user from even launching an Office application. Fortunately, you can run the utility from the Run dialog box with the following entry:

```
drive:\setup.exe /focums install.msi
```

User-level prevention is a magic cure

Every support pro knows that most of their problems could be avoided if their users would just follow a few simple, common sense precautions. If you keep getting calls on corrupted Excel documents, pass around this list of Microsoft-endorsed tips to help users keep their data safe and make your days a little less stressful:

- Move radios, fax machines, televisions, stereos, speakers, and other sources of electromagnetic radiation away from computers.
- If your users are still using floppies, make sure they are not putting them on top of their monitors.
- Regularly run Norton AntiVirus or another utility to scan for viruses.
- Store floppy disks in a disk storage container.
- Don't be a slob—prevent dust from building up around your computer.
- And, most important, exit Windows before restarting or shutting down a client system.

Opening multiple windows

One of the most frustrating problems for Excel users is viewing all the information they need, especially because it often lives in different worksheets. Let your users know about this simple way to view multiple sheets, and they'll love you forever.

1. Click on the tab of the worksheet you want to view.
2. Choose New Window from the Window menu.
3. Repeat this process for each of the worksheets you want to display. (Excel opens each sheet in a window that sits on top of the previous one.)
4. To see them all at one time, choose Arrange from the Window menu.
5. Select Tiled, Horizontal, or Vertical.
6. Select the Windows Of Active Workbook check box.
7. Click OK.

Creating custom lists

Here's another timesaving tip to pass on to your users. If they often make use of the same lists of data, you can teach them to create a custom list instead of typing the same information repeatedly.

For all versions of Excel, if the data is already in a worksheet, follow these steps:

1. Select all of the cells containing the information you want to copy.
2. Select Options from the Tools menu.
3. Click the Custom Lists tab.
4. Click Import.

If you want to create a list from scratch in Excel 97, follow these steps:

1. Go to the Tools menu and choose Custom Lists.
2. Click in the List Entries box.
3. Type each item for your list, pressing [Enter] after each item.
4. When you're finished, click OK to create the list.

To create a list from scratch in Excel 2000/XP, follow these steps:

1. Select Options from the Tools menu.
2. Click the Custom Lists tab.
3. Type each item in the List Entries box, pressing [Enter] after each one.
4. Click Add.

To use a custom list, type the first entry of the list. Use the fill handle at the bottom right corner of the cell to drag the whole list across adjacent cells.

Copying styles between workbooks

Creating formatting styles is a real timesaving trick for users. So why should they waste time re-creating these styles in multiple workbooks? If they have formatted a workbook in a particularly useful way and want to continue to use the format in other workbooks, teach them how to copy the style from one workbook to another:

1. Open the source workbook and the destination workbook.
2. Make sure that the destination workbook is active, go to the Format menu, and choose Style.
3. Click the Merge button.
4. Within the Merge Styles dialog box, select the source workbook from the Merge Styles From list box.
5. Click OK.

A macro that pulls source data from charts

When building charts in Excel 97, many users reference information in other workbooks to keep spreadsheets fairly streamlined. But what do you do if a user complains that the data source has become corrupted or that the file is no longer available in the network?

It's a little bit of a headache to read, but Knowledge Base article 137016 includes the full contents of a macro that can strip important data from the graphic elements of a chart.

<http://support.microsoft.com/support/kb/ARTICLES/Q137/0/16.asp>

After creating the simple VBA macro, you only need to run it and drop the information into a worksheet called ChartData. Don't go volunteering this information to all of your users, but if a VP loses an important sales worksheet, you can pull out this nifty power trick and save the day.

Update for Excel 2000 text export

Microsoft has released a patch that fixes a formatting problem when exporting Excel 2000 files to .txt documents. On machines running a Microsoft OS earlier than Windows 2000, four-digit year entries are automatically truncated to two-digit entries, regardless of formatting applied within Excel.

This behavior occurs only when users employ a macro or procedure to export to a text file, not during manual saves to a text file format. If your users are using Visual Basic for Applications to export files to .txt, .prn, .csv, or .dif formats (and you are using a pre-Windows 2000 OS), you need to install an [update](#) released by Microsoft. For more information, read the Microsoft Knowledge Base article 247796, XL2000: Programmatically Exporting Text File Truncates Dates:

<http://support.microsoft.com/support/kb/articles/Q247/7/96.asp>

Creating a workspace in Excel

Often, your users will be working with several workbooks at a time. They will be happy to know that they can arrange the workbooks, save them as a workspace, and then open everything at once in the viewing configuration they prefer. Here's how:

1. With all the workbooks in the preferred positions, go to the File menu.
2. Choose Save Workspace.
3. Type a name for the file.
4. Click Save.

The next time your users want to work with those workbooks, they can go to File | Open and select their workspace.

Printing formulas

If your users have invested time in writing numerous formulas, they may want to print them for safekeeping. Although Excel doesn't include a simple "Print Formulas" button, functionality is built in. Here's how to find it:

1. Go to the Tools menu.
2. Click Options.
3. Within the section called Window Options on the View tab, select Formulas.
4. Click OK.

There is also a keyboard shortcut for this function. Press [Ctrl]~. In both cases, the result is that the formulas, rather than their results, will appear in the worksheet.

Shifting a scatter chart's axis

In most cases, Excel's default behavior of making the x-axis and y-axis of scatter charts intersect at 0 is desirable. But sometimes, your users may want to change that intersection point. An obvious example is when a user needs to highlight just a selected number of years of data but doesn't want to create an entirely new chart.

To reset an axis, follow these steps:

1. Activate the chart for editing.
2. Click to select the axis you want to modify.
3. Choose Selected Axis from the Format menu.
4. Select the Scale tab.
5. Type a number in the Value (X or Y) Axis Crosses At text box.
6. Click OK.

Recovering data with the SYLK format

If a user complains of being unable to save a workbook, the problem is usually user error. However, things do go wrong. If you've tried all the common sense remedies and the workbook does seem to be corrupted, you can try to revert to the old Symbolic Link (SYLK) format.

This approach will let your users recover most of their essential data by stripping away the more complex information that tends to be the source of corruption. Unfortunately, this trick won't help them get back any data, including charts and graphics, that wasn't supported back in the days of Excel 2.0.

Follow these easy steps in Excel 97:

1. Open the file that appears to be corrupted.
2. Select Save As from the File menu.
3. In the Save As Type list, select SYLK (Symbolic Link).
4. Click Save.
5. Click OK in the warning dialog box that appears when you save in SYLK format.
6. Close the file.
7. Click No to move on past the warning that the file you are closing is not in the normal Microsoft Excel format. (Clicking Yes will resave the file in the normal Microsoft Excel format.)
8. Select Open from the File menu.
9. Select All Files (*.*) in the Files Of Type drop-down list.
10. Select the SYLK file you saved in step 3 and click Open.
11. Select Save As from the File menu.
12. Select Microsoft Excel Workbook (*.xls) in the Save As Type drop-down list.
13. Click Save.

For additional information, check out Microsoft Knowledge Base article 45557:

<http://support.microsoft.com/support/kb/ARTICLES/Q45/5/57.asp>

Faster access to worksheets

When working with large workbooks in which all of the worksheet tabs are not visible, your users may get frustrated with scrolling to find the sheet they need. Here's a really quick tip to pass on. Right-click on any of the tab-scrolling buttons (to the left of the sheet tabs) and a shortcut menu will pop up. You can then select the tab you want.

Inserting the full file path in headers and footers

Excel 97 doesn't offer you the option of displaying the full file path in worksheet headers and footers. With a little basic VBA editing, you can add this functionality to users' systems.

1. Go to Tools | Macro | Visual Basic Editor.
2. Go to the View menu and select Code.
3. Type the following:

```
Sub PathFooter()  
ActiveSheet.PageSetup  
LeftFooter=ActiveWorkbook  
FullName  
End sub
```

4. Go to File | Close.

Return to your workbook and save the file. You can now run your macro by navigating to Macros and choosing the one named PathFooter.

Exceeding the 30-argument limit

Excel has a 30-argument limit for statistical functions. It's easy to get around, though, if you group some of your arguments within parentheses.

Instead of entering `AVG(A1,A2,A3...,A33)`, you can enter `AVG((A1,A2,A3)A4...,A33)`, and Excel will accept the grouped arguments as a single argument within the formula.

Ask Mr. Excel

Excel power users and support pros: Check out MrExcel.com. Mr. Excel answers one reader's question per week, according to what piques his interest, often including full macros to be copied and put to use.

Because this is a side project for the site's operator, he can't answer every question submitted, but he has provided a message board for other Excel users to exchange information. Other site features include past week's tips, quick answers, book reviews, and the opportunity to find out whether you can answer the monthly challenge.

If you work extensively with Excel and Visual Basic, this may be the one resource you've been looking for.

<http://www.mrexcel.com>

Catching data entry errors

It's a fact: Shared workbooks generate more errors than workbooks with only one owner. Here's one way to flag errors as they occur:

1. Select all cells in which data will be entered.
2. Go to Format | Conditional Formatting.
3. In the Conditional Formatting dialog box, specify which cell values to flag.
4. Click Format.
5. In the Format Cells dialog box, click the Patterns tab and select a color to signal errors.
6. Click OK.

Now, when someone enters an error, it will be flagged with your selected color.

Quickly setting a print area

Your users probably waste a lot of time and paper sending entire worksheets to the printer when all they need is a printout of a few columns or rows. Teach them this quick trick to set specific print areas within the Print Preview mode:

1. Select View | Page Break Preview.
2. Select the cells you want to print.
3. Right-click on the selection and choose Set Print Area from the shortcut menu.
4. Click Print.

Counting the days via cell subtraction

By default, Excel 97 adjusts the display of dates to friendly displays that users can identify. But behind the scenes, Excel still thinks in numbers. So finding the number of days between two days is as easy as simply subtracting one cell that contains a date, regardless of format, from another date-carrying cell.

For example, let's suppose a user has entered the date 2/20/2000 in cell A1 and 3/10/2000 in cell A2. The formula A2-A1 will return the result 19 (2000 was a leap year). Note: You must use the date that falls later in the calendar year as the first argument of the equation, or the equation will return an error.

By default, Excel will format the result of a formula in the same format as the source cells. So in our example, it reports that the difference between 2/20/2000 and 3/10/2000 is 1/19/1900. However, simply right-clicking the formula cell, selecting Format Cells, and selecting a Number format will fix this problem.

Quickly shuffling toolbar buttons

Here's a quick tip you can pass on to your users who like to modify their Excel toolbars.

Instead of right-clicking a toolbar and selecting Customize to launch the Customize dialog box, they can simply press the [Alt] key and click and drag a button to any toolbar they want. Of course, both the source and target toolbars must be visible for this procedure to be effective.

One quick word of warning: This technique moves buttons, so they will no longer be on the default toolbar. If users want to copy a button to a new toolbar while retaining its position on the original one, they can hold down [Ctrl][Alt] while clicking and dragging the button.

Opening a workbook at startup

If some of your users spend much of their day working in a specific workbook, you can teach them how to designate that workbook to open each time Excel launches. All they need to do is place the workbook (or a shortcut to the workbook) in the XLStart folder. This technique also works with network shortcuts.

You'll find the XLStart folder in the same place as the Excel program files—typically C:\Program Files\Microsoft Office\Office. After placing the file or shortcut in the XLStart folder, the user can just restart Excel to see the file pop open automatically.

Hiding duplicate records

One of the most common mistakes users make in preparing a report or sorting data is including redundant entries. For example, if a worksheet keeps a running tally of purchases made by Customer 1, there's no need to report cells other than the ones keeping the composite figures. A built-in filter in Excel will take care of this problem.

Follow these steps:

1. Choose Data | Filter | Advanced Filter.
2. Drag across the worksheet to select the lists containing duplicate entries.
3. Select Unique Records Only.
4. Click OK.

Excel will now hide any duplicate records in the selected range.

Converting Julian dates

Here's a little piece of post-Y2K calendar trivia for you. Since the 1950s, astronomers and geophysicists have used Julian dates to track time in continuous, absolute terms, without bothersome seasons, leap years, or man-made conventions. Julian dates associate a date with the number of days elapsed since January 1 of the same year. (For example, Jan. 2, 2000, has the value of 2.)

Excel doesn't have a built-in option to convert dates to Julian time, but you can use this shortcut to convert today's date to the Julian format:

Just type this equation into any cell in your worksheet:

```
=VALUE(RIGHT(YEAR(TODAY()),2)&TEXT(TODAY()-DATE(YEAR(TODAY())-1,12,31),"000"))
```

Now, you'll need to format this value to display five digits.

1. Right-click in the cell that contains the value and select Format Cells.
2. On the Number tab, choose Custom from the Category list box.
3. In the Type text box, enter *00000*.
4. Click OK.

Saving time with Custom Views

Custom Views is a tool in Excel that lets you assign a name to a particular sheet layout so you can recall it for later viewing. You can access the feature by selecting View | Custom Views. In the Custom Views dialog box, you'll see a listing of saved views, and you can add or delete views. When adding a view, you have the option of including print settings, hidden rows and columns, and filter settings. The new view even remembers where the active cell was when the view was saved, so when you open that view, it will "jump" to the spot in your sheet that corresponds to that view.

Making good use of custom views can save a great deal of time. For example, you can eliminate repetitive hiding, resizing, filtering, and other changes for producing various printed reports. Each set of options can be saved as a view. Then, all you need to do is apply the view before you print.

Beware of sharing dates between Excel for Windows and Excel for Macs

Excel supports two data systems: 1900 and 1904. Early Macintoshes did not support dates before Jan. 1, 1904. For this reason, Excel for Macs renders entered dates as the number of days since 1/1/1904. Excel for Windows, on the other hand, supports dates starting Jan. 1, 1900, and so renders dates as the number of days since 1/1/1900.

For example, the date 1/1/1904 is represented as a 0 in the 1904 date system. The same date in the 1900 date system is represented by the number 1,462. This is because 1/1/1904 is 1,462 days since 1/1/1900, the day the 1900 date system begins its "counting."

What all this means to you is that if you have users on both Macs and PCs using Excel and copying data between workbooks, there's the chance of incorrect date representation.

One solution is to set your PCs so that they all use the 1904 date system. Then they'll be compatible with the Macs. To do so, open a workbook and select Tools | Options. Click the Calculation tab. Under Workbook Option, select the 1904 Date System check box and click OK. For more information, check out Microsoft's Knowledge Base article 214330, which covers this issue in depth:

<http://support.microsoft.com/support/kb/articles/Q214/3/30.asp>

Don't use AutoFormat on an entire worksheet

Microsoft has confirmed that there's a potential problem when AutoFormat is selected for a whole worksheet rather than a specific range.

Depending on your OS and configuration, you could see symptoms ranging from Excel not responding to various error messages concerning page faults, low virtual memory, and access violations.

To resolve this issue, make sure that you select only the range of cells you want formatted when you select Format | AutoFormat.

Microsoft Knowledge Base article 211478 offers the details of the problem:

<http://support.microsoft.com/support/kb/articles/Q211/4/78.ASP>

Keep track of changes

Track Changes works hand-in-hand with workbook sharing. It keeps a log of all changes to a workbook. It tracks cell changes, worksheet additions and deletions, column changes, and other types of changes, as well as the date, time, and name of the user who made the change. When data is changed, it keeps a record of the old and new values and the exact sheet and cell where the edit was made.

One of the best parts of this tool is the ability to have Excel make a list of the changes in a separate worksheet. When this option is selected, Excel builds a listing of each change and even turns on AutoFilter to make finding specific changes easy. It also allows you to review each change and either accept or reject it. Rejecting a change restores the original value.

To turn on this tool, just select Tools | Track Changes | Highlight Changes. The Highlight Changes dialog box provides options such as tracking who makes the change, when the change is made, highlighting the change on-screen, and listing the changes.

When using this feature, you might like to see the Summary sheet when you open your workbook. Excel removes it when you save the sheet, but you can use this macro to have Excel quickly rebuild it:

```
Sub ViewChangeSummary()  
With ActiveWorkbook  
HighlightChangesOptions When:=xlAllChanges  
ListChangesOnNewSheet = True  
HighlightChangesOnScreen = True  
End With  
End Sub
```

Conditional formatting

Conditional formatting is one of Excel's better features. It allows you to preset certain font styles, colors, and cell-background colors based on cell values. This can be very useful for highlighting important information and values outside an accepted range or providing a visual cue to associate value ranges with color codes.

The best part is that conditional formatting is easy to set up. Just click the cells you'd like to format and select Format | Conditional Formatting. The Conditional Formatting dialog box lets you set up the conditions by which the formatting of the cell will occur. You pick the operator (between, equal to, less than, etc.) and the value or range of values. Click Format to open the Format Cells dialog box, where you can select the colors and styles to be used.

Each cell can have several conditional formats. For example, you might say that if a certain cell's value is between 20 and 50, the text should be blue on a yellow background. But you can format that same cell to exhibit red, bolded text on a green background if it contains a value between 51 and 100.

Data entry solutions with the Template Wizard

One of the hardest parts of gathering data from users is creating the right collection application. The data collection front end is often a series of forms in Access or ASP pages. However, the setup and security issues associated with these methods are sometimes too much to deal with for smaller data collection needs.

Excel 97 and 2000 ship with an add-in that can help this situation. The Template Wizard with the Data Tracking add-in will help you create an Excel template that will accept data from users in a familiar environment and then transfer the data to an Access database after it's saved. The wizard will walk you through the steps of defining the tables where the data should be inserted.

This is a great solution for such things as purchase order requests, invoices, or expense forms. Your users can enter their data in Excel, and the data structures set up by the Wizard will place that data into the Access database. The data will be collected into one repository for easy reporting and analysis.

Microsoft Knowledge Base article 214246 explains the process in detail:

<http://support.microsoft.com/support/kb/articles/Q214/2/46.ASP>

A separate article explains how to use the wizard with Excel 97:

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;148314>

Excel XP users don't have to forgo this handy add-in, although it doesn't ship with the program. Microsoft has made the wizard for Excel XP [available for download](#).

Customizing "division by 0" error messages

Working in a complex worksheet, you may have encountered that annoying "#DIV/0!" error when the divisor of your formula is a zero. Using the built-in IF function, you can create your own message for display when you divide by zero:

```
=IF (DIVISOR = 0, "Your Text", DIVIDED/DIVISOR)
```

The IF function evaluates the first parameter (DIVISOR = 0). If it's TRUE, it places the second parameter ("Your Text") in the cell. If it's FALSE, the function places the third parameter (DIVIDED/DIVISOR) in the cell.

This helps you control the message that's displayed when your formula produces a zero value that is then used to divide another value.

Debugging your code

A great way to find errors in your code is to step through it one line at a time. The Step Into button on the Debug toolbar in the VBA Editor lets you do this.

To start your macro, just click Step Into on the toolbar. The VBA Editor will work through each statement. This allows you to take advantage of some of the other debug tools, such as the Locals window and the Watchwindow.

When stepping through your code, you can use the Step Out command to run the remaining code as normal. So even if you stepped through half the macro one statement at a time, you can then run the last half normally by clicking Step Out.

Debugging using watches

You've just spent over an hour writing just the macro you need. It's the one that will make your life easier, and you just got it right—or so you thought. But now it's providing the wrong data output, and you can't figure out why. All the code looks right.

One of the great debug tools in the VBA Editor is called a watch expression. Watch expressions allow you to see the values of the variables in your macro at any given time. You can use them with the Step commands to walk through your code. These values can be a big help in finding out why your macro is generating the wrong data. You can use the watch expressions to see your variable values and then compare them to what they would be if your macro were working correctly.

To access the Watch window, just click View | Watch Window in the VBA Editor. To add a watch, right-click a variable and select Add Watch from the shortcut menu. This brings up a dialog box that gives you options about the new watch.

Ensure proper data entry with validation

Data validation rules in Excel automatically check an entry made by a user against rules you have preset in the worksheet. This is helpful when you have formulas that depend on certain expected input from the user. These formulas can often return errors if the format of the entered data is wrong.

To set data validation rules, select Data | Validation. The dialog box allows you to pick a data type to require and a range of values to accept. In the Input Message tab, you can choose to display an instructional message when the cell is selected. This message appears in a pop-up, note-style display. It can be used to provide users with information about what data you expect them to enter. You can also provide a message from the Error Alert tab if the data entered does not meet your rules.

Don't assume that your users will enter the data you expect. If your worksheet functionality depends upon consistent data entry, look into using data validation.

Hyperlink to a specific cell

Did you know that you can create a hyperlink from a Web page on your intranet or the Internet to a particular sheet, range, and cell in an Excel workbook? For example:

```
<a href="http://yourwebserver/hyperlinktesting  
.xls#Sheet1!A1">Link Test </a>
```

will open your file called Hyperlinktesting.xls and then select cell A1 on Sheet1.

Using this kind of linking, you can specify exactly where your user will enter the workbook. Users' views of the workbook won't be bound by the state at the last save.

This can be helpful if you insert the link to a Web page that asks the user to enter data in a particular cell or area of a sheet. You can select the cell for them when they click the link.

For information on linking to Excel 97 worksheets, see Microsoft Knowledge Base article 172503:

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;172503>

For details on linking to Excel 2000 worksheets, see Microsoft Knowledge Base article 211693:

<http://support.microsoft.com/support/kb/articles/q211/6/93.asp>

Macro security settings

If you write macros that other people in your organization use, you should be aware of the Macro Security settings in Excel (and in all Office applications). These settings control the messages you receive when you open a workbook that contains macros; they ask if it's okay to enable them.

In the good old days, developers didn't have to worry about such things. But with the advent of macro viruses, you now need to take such user-level security settings into account when developing code.

In Office 2000/XP, there are three levels of security for macros contained within files:

- **High** automatically disables any macro not "signed" with a digital signature. It will allow macros to be enabled only from sources already deemed to be "trusted."
- **Medium** prompts users to decide whether a macro should be enabled.
- **Low** does not notify users that files contain macros. All macros are enabled for opened files.

The security settings on your users' machines might be dictated by policy or may be up to the user's discretion. Either way, the safest action for developers of Office-based macros is to obtain a digital signature to sign their macros. With a digitally signed macro, even the High setting will enable your code, provided the user approves.

Express yourself with comments

We've all opened the workbook from hell. The formatting was a mess, and there were formulas everywhere. A day later, you may almost have it decoded enough to work on it.

Get your users into the habit of documenting workbooks with comments. They can enter text descriptions into any cell. These descriptions are visible only if you hover the mouse over the cell. A cell with a comment is marked with a small red triangle in the upper-right corner. Your users can easily add a comment to a cell by clicking Insert | Comment.

Excel's Reviewing toolbar facilitates working with the comments in a workbook. The options on this toolbar allow users to add and delete comments and move from one comment to another in a workbook.

With the growing use of teams and collaboration, it's more important than ever to ensure that your users' intentions are clearly communicated to everyone who needs to use a workbook. Comments are a good way to start.

Map your data—on real maps

If you often analyze geographically based data in Excel, you should check out Microsoft MapPoint 2000. MapPoint lets you create customized maps, and you can arrange your data on a map to suit your needs. For example, MapPoint can show average household income demographic data (which is built into MapPoint) and then show your company sales data for the same regions. This can help you see trends in how your organization sells across demographic groups. You can save these maps, annotate them, and use them later.

If your organization tracks anything by region, city, etc., take a look at MapPoint. It's also a great general-purpose mapping application that will let you create maps for your Windows CE devices using Microsoft Pocket Streets. For more on MapPoint, visit <http://www.microsoft.com/office/mappoint/>.

Create an Outlook message from Excel

The ability to create objects from other Office applications is one of the great things about using VBA. For example, you can write a macro that will create an Outlook message object and attach a shortcut to the currently open workbook. A button on the toolbar will perform this same action, which is fine if that's all you want to do. But using the sample macro below, you can add other text to the message, pre-address it, or perform other actions. You can also tie the creation of the message to some other user action. For instance, you could automatically send a message to certain people if the user saves the workbook and if

the values of certain cells match criteria you set—all without any user interaction. This sample macro will get you started in the right direction:

```
Sub CreateOutlookMail()  
Dim olApp As Outlook.Application  
Dim olMailMessage As Outlook.MailItem  
  
' Create new instance of Outlook or open current instance.  
Set olApp = New Outlook.Application  
' Create new message.  
Set olMailMessage = olApp.CreateItem(olMailItem)  
With olMailMessage  
    .Subject = "Testing mail by Automation"  
    .Body = "This message was created by VBA code running Outlook through Automation." _  
    & Chr(13)  
    .Attachments.Add Application.ActiveWorkbook.Path & "\" & _  
    Application.ActiveWorkbook.Name, olByReference  
  
    .Display  
End With  
Set olMailMessage = Nothing  
olApp.Quit  
Set olApp = Nothing  
End Sub
```

Repairing broken file type associations

How many times has one of your users mysteriously lost his or her file associations, requiring you to go to the File Types tab and re-associate Excel (or Word and PowerPoint) to the .xls and .xlt file types?

There's a faster way to fix Office file type associations, and it takes just one quick step. Using the `/regserver` command line switch will make Excel reregister itself and rewrite all its registry entries, including its file type associations. You can reset all your users' file extensions for Excel, Word, PowerPoint, and Access using this same method.

From the command line, type `C:\Program Files\Microsoft Office\Office\Excel.exe /regserver`. You can also modify a shortcut to Excel by adding `/regserver` after the end quotes in the Target field of the Shortcut tab on the shortcut's Properties sheet.

Web Publishing Wizard Error with Office 2000

When trying to run the Setup utility after your initial Office 2000 setup, you may receive the following error:

```
Insert Disk  
The file 'Crswpp.dll' on Web Publishing Wizard disk cannot be found.  
Insert Web Publishing Wizard disk in the selected drive, and click OK.
```

Further efforts to run Setup will produce similar errors.

Microsoft has identified the problem as being connected to the installation of some modem hardware. Fortunately, clicking Cancel when you receive this error message will usually prompt Setup to run successfully with no harmful effects.

For more information about all the causal factors, including a complete description of the problematic modem hardware and possible resolutions, check out Microsoft Knowledge Base article 234688:

<http://support.microsoft.com/support/kb/articles/Q234/6/88.asp>

Use auditing to troubleshoot

We've all clicked a cell to see the formula, wanting to find out where the result came from, only to find a formula that went on forever. It has everything: nested functions inside of conditionals, pulling data from cells all over the workbook. Just when you think you'll never figure this monster out, you remember Excel's Auditing features.

Select the cell containing the complex formula and click Tools | Auditing | Trace Precedents. (In Excel XP, choose Tools | Formula Auditing | Trace Precedents.) Excel will draw blue arrows from every cell the formula references to the cell containing the formula. Now you can see where that data is coming from.

Auditing also allows you to select a cell containing a value and choose Trace Dependents, which will draw arrows from the cell to all the cells containing formulas that reference that value.

And if you have a cell that shows an error value, you can select it and choose Trace Error. Excel will show you all the cells that contribute to the error.

Using Goal Seek

You have a worksheet that has the price of a house, closing costs, and interest rate, and it calculates your monthly payment. It works great, but now you want to enter a desirable payment and find the price you can afford to get that payment.

This is a common situation when dealing with worksheets (although usually more complex than a simple mortgage formula), especially when dealing with projections or forecasts. Goal Seek is just what you need for this situation. It allows you to specify a target value for a cell and tell Excel to change the value of another related cell to obtain this value.

Highlight the cell you want to change and click Tools | Goal Seek. The cell reference for the selected cell will appear in the Set Cell field. Choose a target value and enter it in the To Value field, then specify what cell should be changed to meet your target. When you click OK, Excel will attempt to meet your target.

Using shared workbooks

If you have several users who need to work from the same workbook at the same time, familiarize yourself with Excel's Shared Workbook feature. This feature can be a lifesaver when workbooks must be updated by multiple users.

To use this feature, select Tools | Share Workbook, and Excel will allow multiple users to edit the workbook. You can set options that control when changes are updated and how to handle conflicting edits by different users. You can also choose to save updates when a user clicks Save, or you can specify a time interval for Excel to automatically save user updates. In the Share Workbook dialog box, you can decide how changes should be logged by the Track Changes feature, which integrates with workbook sharing. That way, you can see exactly what changes were made to cells, when sheets were added or deleted, how conflicting edits were resolved, etc. Users can also see the names of all the other users who have the workbook open for editing.

Use the Immediate window

One of the tools provided by the VBA Editor is the Immediate window. You can use it as a general-purpose scratch pad while testing your code. For instance, you can place Print commands at various places in your code to output variable values or the results of calculations to this window. For example:

```
For Count = 1 to 20
    Debug.Print Count
Next Count
```

This code prints the numbers 1 through 20 to the Immediate window.

You can use the Immediate window to check or change variable values or object properties to test the results on your running code. You can also call other procedures from this window. For example:

```
Debug.Print A * 125
```

multiplies the current value of the variable *A* by 125 and prints the results in the window.

```
Activeworksheet.Name = "New Name"
```

changes the name of the active sheet.

The Immediate window offers an excellent way to test the outcome of new values or property changes on your code without inserting the new code into your procedure.

Summarize data with grouping

Grouping in Excel allows you to create outline structures with your data. Let's say that you have a column of numbers with a cell at the bottom that contains their sum. You could just leave it at that and have other worksheet users view the entire column every time they open the workbook. Or you could use Grouping to hide the numbers, showing only the sum but providing the option of expanding the outline to see the hidden data.

Grouping adds a small column to the left of your data with plus and minus symbols to let users expand or hide the detailed data. To build this outline, select the columns of data that will be hidden in your outline and choose Data | Group And Outline | Group. Excel will add the expand/collapse symbol to the left of your data. Your users can now choose whether to see the source data for the calculations or just the totals.

You can choose whether the outline should assume the Totals row is above or below the data. You can also use this feature to summarize rows in the same way, so an outline expands or collapses across rows instead of columns.

Take a few shortcuts

Who doesn't need to save a little time here and there? Keyboard shortcuts can help you do just that. Here's a list of shortcuts for Excel that will make life a little easier:

- [Ctrl][Shift]A: Use this combination after typing a function name, and it will automatically enter the parentheses and the argument names for the function you typed.
- [Alt][F8]: Displays the Macro dialog box.
- [Alt][F11]: Opens the VBA Editor or switches to it if it's already open.
- [Ctrl]1: Displays the Format Cells dialog box.
- [Ctrl][Shift][~]: Applies the General format.
- [Ctrl][Shift][\$]: Formats the current cells as Currency.
- [Ctrl][Shift][#]: Applies the Date format.
- [Shift][F10]: Displays the shortcut menu that applies to the selection.
- [F6]: Moves to the next pane in a split worksheet.
- [Shift][F6]: Moves to the previous pane.
- [Ctrl][F6]: Moves to the next workbook window.
- [Ctrl][Shift][F6]: Moves back to the last workbook window.
- [F3]: Opens a dialog box to paste a Named Range into a formula.
- [Ctrl][F3]: Opens the Define Name dialog box.
- [Ctrl][Shift][F3]: Automatically creates Named Ranges from the headers for the selected table of data with row or column headers.
- [Shift][F3]: Inserts an equal sign (=) and opens the Paste Function dialog box to help you pick a formula.
- [Alt][=]: Inserts the AutoSum function.
- [Ctrl][:]: Inserts the current date.

Analyzing external data in Excel

Your users might not know that they can use Excel to analyze data from the various databases in your organization. They can use the powerful analytical tools of Excel on data they previously only viewed or edited within the database user interface. Using these features, they can pull data from your customer database, sales tracking database, product catalog, and HR systems into various sheets in a workbook and perform complex pivot table analysis not possible using the separate applications.

This ability is provided by the use of Microsoft Query, an Office application that lives "behind" Excel. You can access Microsoft Query by clicking Data | Get External Data. The Query interface is similar to the Design view of a query in Access. Point Query at a database, and it will allow you to build complex relational queries to pull the necessary data into your worksheet. Now you can build your pivot tables and perform your analysis.

Take advantage of the Personal Macro Workbook

The Personal Macro Workbook is the Excel cousin to the Normal.dot file in Word. It's a file that opens with Excel in the background and makes your saved macros available for any open workbook. This workbook is one of the options available when recording new macros in Excel.

We recommend saving all macros that will be used across multiple workbooks into the Personal Macro Workbook. This will keep things simpler for your users. If they try to save a macro into each workbook where it will be used, they'll have a tough job of updating if the macro requires a change. If it's saved in the Personal Macro Workbook, it will need to be changed only once. The change will be available across all workbooks. When writing macros in the VBA Editor, macros should be saved in Personal.xls.

Remember to retain this workbook when doing backups. It can be a valuable file once you begin creating macros that help in your work.

Clear out all comments

As handy as comments can be in annotating the information in a worksheet, your users may sometimes want to delete them once they've served their purpose. The fastest way to handle this task is to click Edit | Go To | Special and then select the Comments option and click OK. Users can then choose Clear from the Edit menu and select Comments. Excel will delete all the comments and their text from the current worksheet.