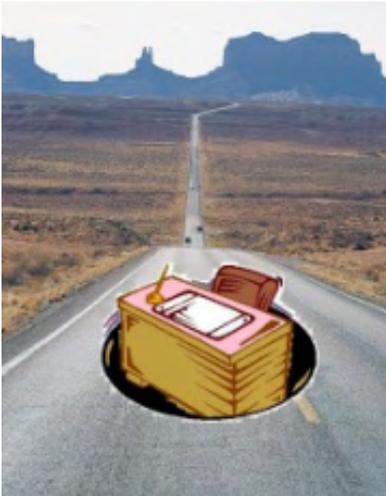


ComputerEdge™ Online — 07/10/09



This issue: The Free OpenOffice.org Software Suite

OpenOffice.org (OO.o) has become a threat to Microsoft Office because it's a good program for most applications—plus it's free. Here's what it is, how to set it up, and how to create OO.o macros for saving time.

Table of Contents:

[Digital Dave](#) by *Digital Dave*

Digital Dave answers your tech questions. How secure is instant messaging from prying eyes?; a reader wants to open his System Configuration screen in Vista in maximized mode; a reader wants to upgrade to Win7 without installing Vista.

[The Basics of OpenOffice.org](#) by Michael J. Ross

Can it effectively replace Microsoft Office? The inevitable backlash against Microsoft Office has taken the form of alternative office suites and individual programs, such as word processors and spreadsheet applications. The most well-known, all-encompassing package is OpenOffice.org. What does it have to offer?

[OpenOffice.org Writer Macros](#) by Michael J. Ross

Take the tedium out of repetitive editing tasks. OpenOffice.org, the free and powerful office productivity suite that is gaining fans worldwide, supports the creation and use of macros, which are the best way to take the tedium out of repetitive editing tasks.

[Windows Vista and Windows 7 Tips and Tricks](#) by Jack Dunning

Windows 7 Upgrade Deal Ends July 11! Does it really matter? Microsoft is anxious for you to pre-order Windows 7 upgrades. Should you rush out and purchase it? Also, more on problems with the Internet Explorer 8 update, and a tip on easier file-saving in Windows 7.

(Click Banner)

(Click Banner)

(Click Banner)

[Wally Wang's Apple Farm](#) by Wally Wang

The Best Software Is Often Free

In case you'd rather not pay the upgrade fee for the iLife suite, or simply find that iLife doesn't meet your needs, here are some alternative programs that are completely free. Also, a look at Microsoft's scatterbrained marketing campaign, and a tip on using Dashboard Web Clips.

[Little Linux Lessons: Tips and Tricks from Users](#) by ComputerEdge Staff

Linux users share ideas and ask for help.

A reader writes in with a tip on Compiz, a program that runs cool desktop effects in Ubuntu Linux.

[ComputerTutor Does Microsoft Access](#) by Rob Spahitz

Troubleshooting: Name Parsing

Last week, we created a procedure to allow a user to enter a full name and let us split it into a first and last name to put it into the corresponding fields. It didn't quite work as expected, so we'll explore ways to help us learn how to deal with such issues.

[Programmer's Corner: Runtime Revolution](#) by Wally Wang

Part II: Creating a User Interface in Runtime Revolution

Like most RAD tools, creating a user interface in Runtime Revolution involves a two-step process. First, you have to place a user-interface object on a window. Second, you have to customize that object, changing its properties.

[ComputerQuick Reviews](#) by ComputerEdge Staff

Send us your thoughts about your favorite hardware, software and Web sites.

We want readers' opinions about hardware, software, and Web sites. Share your thoughts and ideas!

DEPARTMENTS:

[EdgeWord: Technology to Change the Way You See Things](#) by Jack

Dunning

Handheld devices are evolving to save battery life.

The screen-power problem is being addressed by new technologies, such as IMOD, MEMS and OLED.

[Editor's Letters: Tips and Thoughts from Readers](#) by ComputerEdge Staff

Computer and Internet tips, plus comments on the articles and columns.

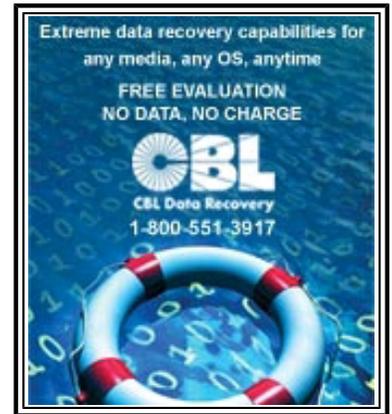
"Access and FoxPro?," "A-Squared," "Using Windows 7," "Checking E-Mail for Scams," "Great Windows 7 Article!"



(Click Banner)



(Click Banner)



(Click Banner)



(Click Banner)



(Click Banner)

[Return to Table of Contents](#)



Digital Dave

“Digital Dave answers your tech questions.” by *Digital Dave*

How secure is instant messaging from prying eyes?; a reader wants to open his System Configuration screen in Vista in maximized mode; a reader wants to upgrade to Win7 without installing Vista.

Dear Digital Dave,

How secure is instant messaging from prying eyes? Also, how safe is it from viruses? Do you have any suggestions as to the best program to use?

*Grace
Denver, CO*

Dear Grace,

Using an instant messenger (IM) is as safe as using e-mail, which means it's *not* secure and *is* susceptible to viruses. That doesn't mean that you shouldn't use it—just don't send out any information of a confidential nature and don't click on unknown links.

While using an IM is like a private conversation between two people, the messages go out over the Internet, passing through a number of servers. The messages could be intercepted and read anywhere along the way. The vast majority of people will never have access to your conversations, but those people who are filtering Internet traffic are looking for credit card numbers, Social Security numbers, and other informational items worth stealing. The government is mostly looking for terrorists.

The best way to protect yourself is to keep your conversations dead boring. Don't put any financial information in, nor should you spice up your conversation. Most filters look for specific words or patterns. Keep it clean.

Honestly, even if you were a little careless in your IMs, the volume of traffic is so high on the Internet that the odds of someone getting a hold of your conversation are very remote. But it's still better to be safe.

This is also true for e-mail. The number-one type of virus attacks that I've seen for IMs are Web links that appears to be from one of your friends. If someone sends you a link, don't click it until you verify that it came from that person. The major IM services have worked hard to plug the holes in their systems that allow viruses to spread. In most cases, you won't see a problem.

Digital Dave

Dear Digital Dave,

I often refer to the System Configuration screen in my Vista OS. There's a lot of information there, but it always opens in a small screen so that you have to move columns and scroll bars around to see everything. Is there some way to get it to open in a maximum window?

*David Posner
San Diego, CA*

Dear David,

As far as I can determine, there is no way to maximize the System Configuration window in Windows Vista.

The System Configuration window is designed for a quick look at your system—in addition to temporarily changing settings while troubleshooting problems. There are two tabs in the window—Services and Startup—that require scrolling. The Services tab only allows you to enable and disable those items. The Startup tab has the same two options.

In most cases, you will want to use the Services program in Windows Vista to work with the items seen in the Services tab of the System Configuration window. Not only can you resize the window in the Services program but, if you right-click a service and select Properties, you will get many more options for both startup and running. If you need to work with startup programs that are not run as services, then Software Explorer in Windows Defender will be useful.

Digital Dave

Dear Digital Dave,

When I bought my last (Dell) computer, Vista was available but I ordered the machine with XP installed. I did pay for the Vista Home Premium upgrade and do have the DVD. I never did install it, as I had a bad feeling that it was another MS fiasco like Windows Me.

I have ordered an upgrade to Windows 7 Home Premium. Will I be able to do a "Clean" install using just the key code on my Vista Home Premium upgrade DVD, or will I actually have to install my Vista upgrade to be able to upgrade from there to Windows 7?

Steve

Escondido, CA

Dear Steve,

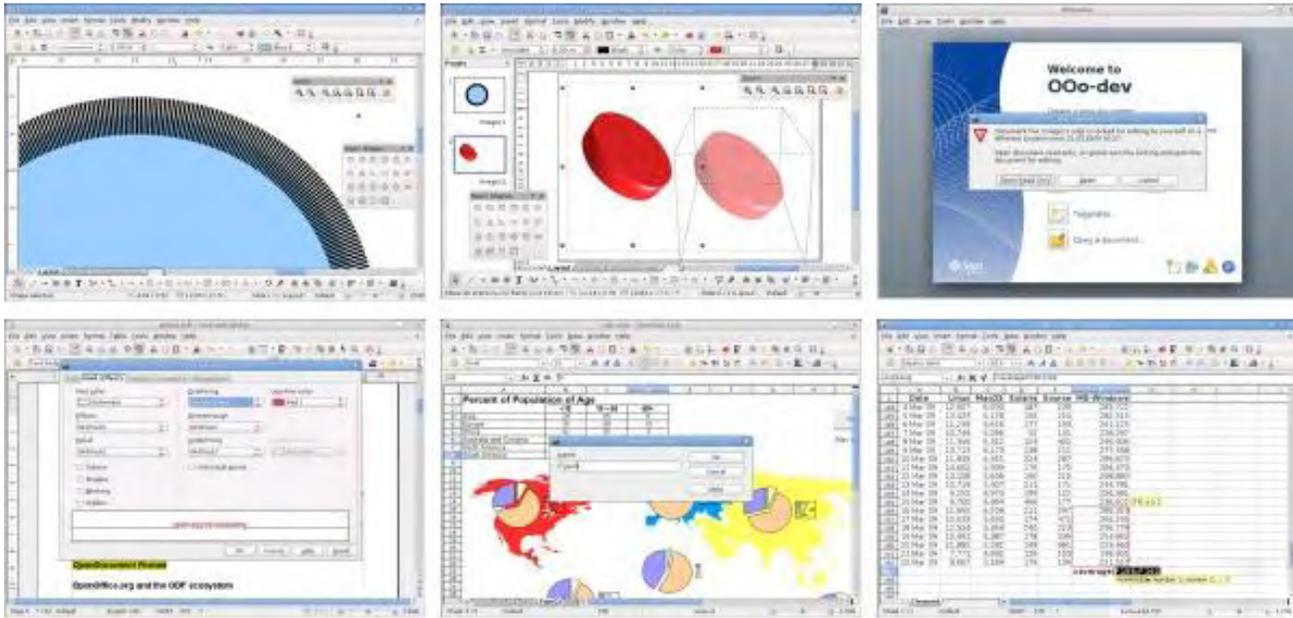
No, you will not need to upgrade your XP machine to Windows Vista before upgrading to Windows 7. You will be able to do a "Clean" install from XP to Windows 7, which means that you will be starting all over again. You will need to save all your files to another storage media and reinstall all of your programs.

My understanding is that the upgrade version of Windows 7 looks for a copy of either XP or Vista on the computer. That is the extent of the checking. You should not need the Vista code key.

Windows 7 Upgrade will perform an actual upgrade (keeping programs and files intact) only if you're moving up from Windows Vista. Otherwise (as is the case with XP), a totally new (Clean) install is done. If you want to do the upgrade without needing to reinstall your files and programs, you must first upgrade to Vista, then to Windows 7.

Digital Dave

[Return to Table of Contents](#)



The Basics of OpenOffice.org

“Can it effectively replace Microsoft Office?” by Michael J. Ross

The inevitable backlash against Microsoft Office has taken the form of alternative office suites and individual programs, such as word processors and spreadsheet applications. The most well-known, all-encompassing package is OpenOffice.org. What does it have to offer?

Typical computer users may utilize a number of applications on a daily basis, but the most critical software—aside from a Web browser—may be an office-productivity suite, usually consisting of related programs for creating documents, spreadsheets, presentations and, in some cases, databases and drawings.



"James decided to create his own version of OpenOffice.org."

Microsoft Office.

Yet many users and industry pundits frequently pointed to the superior features of the products that had fallen by the wayside. They also decried how, with each of the nine major releases of Office (versions 3.0 through 2007), the package has demanded ever-increasing licensing costs and consumption of disk space and system memory, all to deliver features that are rarely if ever used.

Even though much of this functionality is gradually migrating from the desktop to the Web (witness Google Docs (docs.google.com/)), most of that work is still being accomplished locally, on personal computers that don't have to be connected to the Internet. This is especially true for professional work done inside corporations or other organizations—the majority of which do not yet trust "the cloud" (Web storage) for safeguarding their business-sensitive data.

In the realm of office-productivity suites, the dominant player has long been Microsoft Office (office.microsoft.com/en-us/products), which can trace its origins to the early 1990s, during which time it battled—and eventually vanquished—competitors such as WordPerfect and Lotus 1-2-3. By the late 1990s, the marketing juggernaut from Redmond had prevailed, and the term "office productivity suite" had effectively become synonymous with

The inevitable backlash has taken the form of alternative office suites and individual programs, such as word processors and spreadsheet applications. The most well-known, all-encompassing package is OpenOffice.org, which bills itself as "the leading open-source office software suite for word processing, spreadsheets, presentations, graphics, databases and more." On the surface, it certainly possesses some appealing advantages over Microsoft's offering. For one, it is completely free, versus the sticker price for Office, which ranges from roughly \$150 up to \$680, depending upon the particular edition.

In terms of functionality, fans of OpenOffice.org contend that it possesses all of the features that people normally need in an office productivity suite, without the unnecessary complexity, security problems, and Microsoft lock-in of its better-known rival. In addition, it possesses almost all of the features found in its non-free counterpart, StarOffice (www.sun.com/software/staroffice/), which is distributed by Sun Microsystems (now part of Oracle).

To get a better idea of what OpenOffice.org has to offer, let's take it for a test spin. We will be using version 3.0.1, the latest as of this writing.

Open Your Office

The first step in trying out OpenOffice.org is to visit its home page (www.openoffice.org/), and click on the download section, indicated by the green arrow.



Figure 1. OpenOffice.org home page.

When the installation file has finished downloading, open the file, and then step through the installation dialog boxes. The default selections should work fine for most if not all users.

After the installation process is complete, you can start OpenOffice.org by selecting it from your Windows Start menu—or the icon on your desktop, assuming that you did not decline to have it added there during installation.



Figure 2. OpenOffice.org welcome dialog.

The welcome dialog boxes will ask you for your name and initials, whether you want the program to automatically check the OpenOffice.org Web site for updates (none of your personal information is transferred to the site), and whether you would like to register (which helps support OpenOffice.org by increasing the number of registered users it can claim).



Figure 3. OpenOffice.org welcome screen.

After the program has started, you are shown a welcome screen, at which point you can create a new text document, spreadsheet, presentation, drawing, database, or math formula. There is also a button for viewing all of the existing templates, and another button for opening an existing document.

We will next explore in more detail all six of the major components of the suite, with emphasis on the first two.

Your Online Writer

The word processing program within OpenOffice.org is known as Writer, and it offers everything you would expect from a modern, fully equipped word processor or desktop publisher.

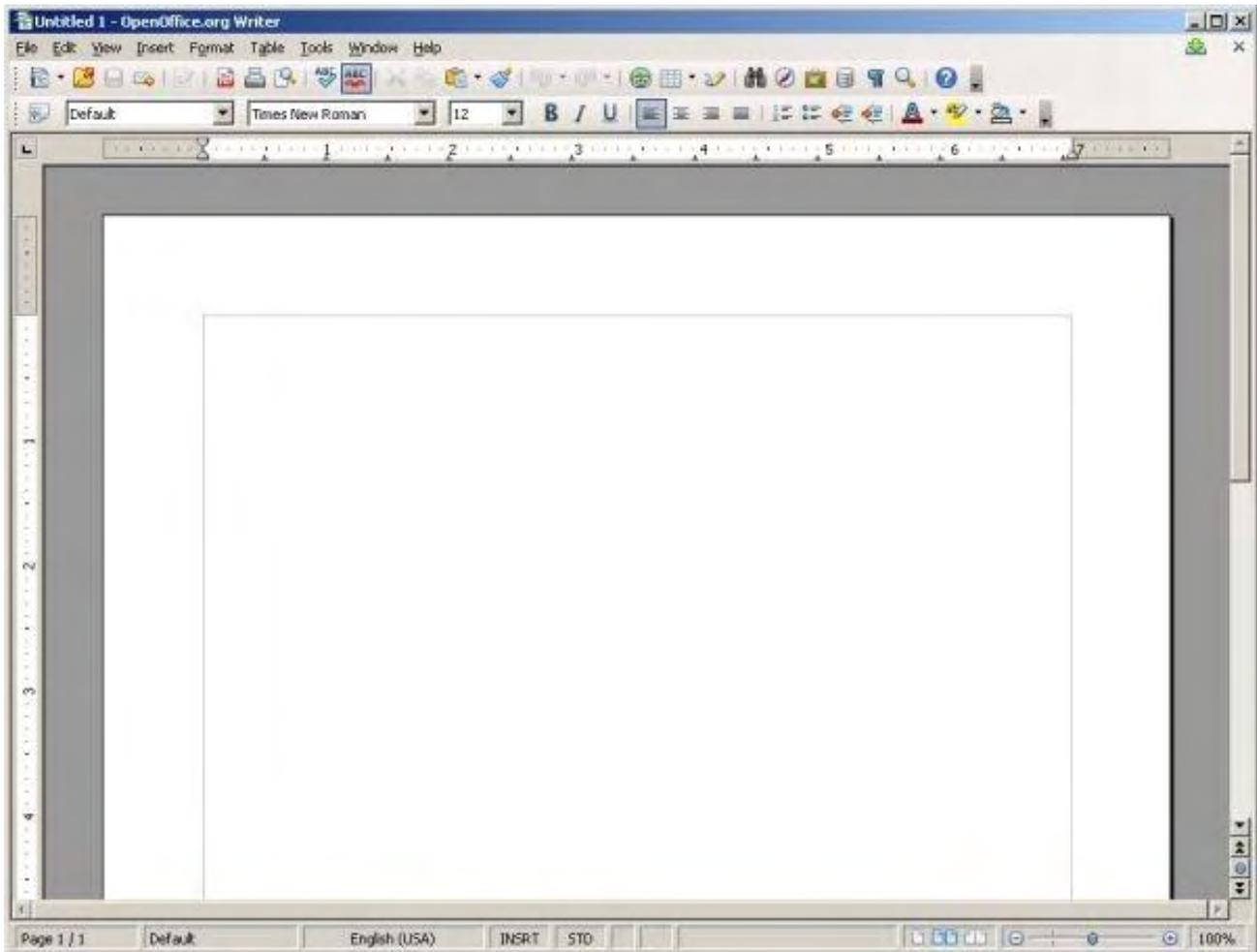


Figure 4. Writer new document.

In so many ways, Writer is quite similar to Office—with all the features that you would expect to find in a major word processor: font faces and sizes, text-formatting options, paragraph styles, tables, spelling and grammar checking, automatic correction and formatting, table of contents, indexing, macros, mail merge, customizable settings, export-as-PDF functionality, online help, and much more.

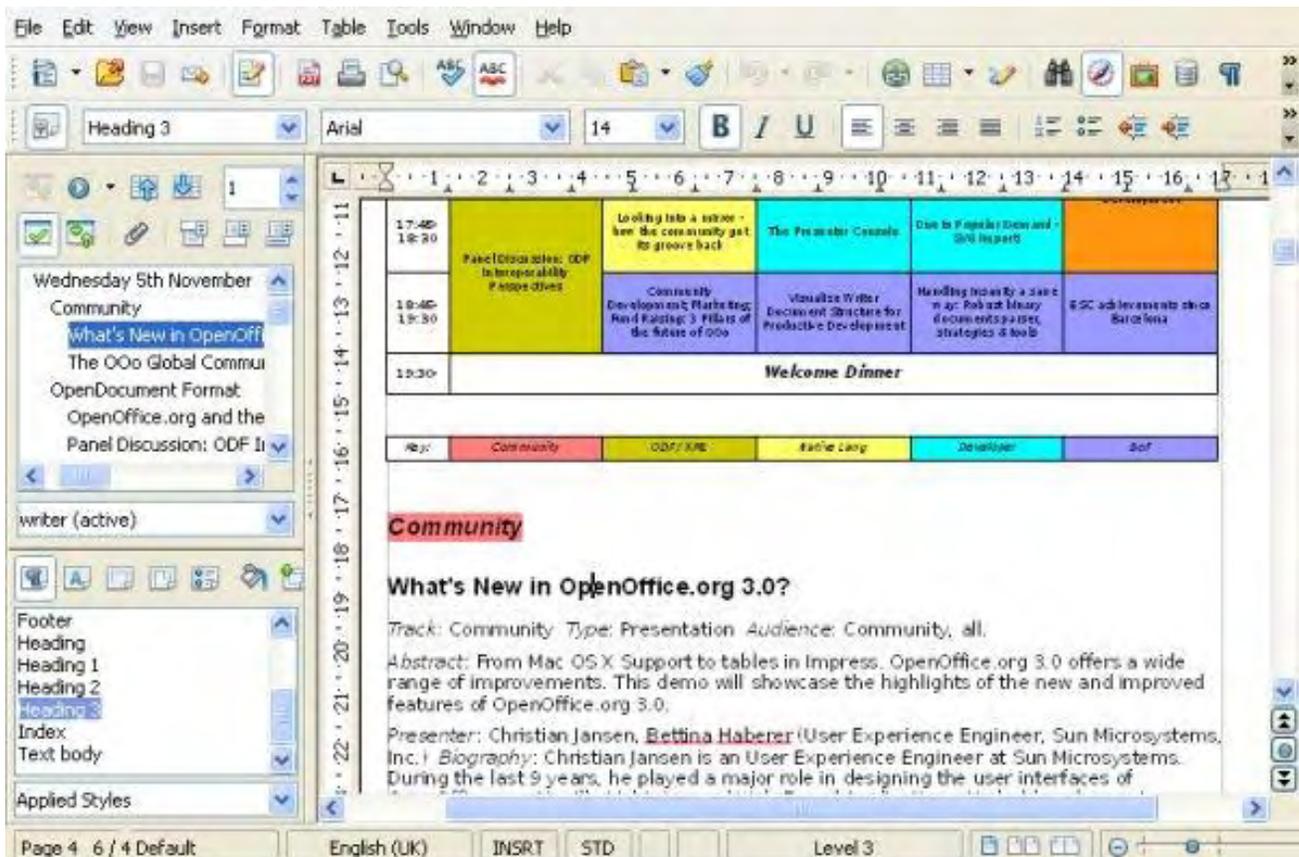


Figure 5. Writer existing document.

For the longest time, Word stored a document's contents in a proprietary binary format, making it quite difficult for users and developers to access the contents programmatically. With Word 2007, Microsoft introduced its own standard (naturally), Office Open XML. In contrast, OpenOffice.org has, from the start, stored all data in open XML formats. In version 1.x, the default file extensions were ".sxw" for documents and ".stw" for templates. Version 3.x utilizes the OpenDocument file format (ODF), an open standard format growing in usage and popularity. Documents can also be saved as HTML, plain text, RTF, and several different versions of StarWriter and Microsoft Word. By the way, all of these XML-based formats are actually ZIP files.

One glaring omission from Writer is the lack of an equivalent to Office's outline view, which can be invaluable for organizing documents in a hierarchical format, and collapsing and expanding sections as needed. Proponents of Writer point to the Navigator panel (Edit/Navigator) as a substitute, because it allows you to promote/demote sections, and move them up and down within the document, without having to cut and paste sections of text within the document. But I found it to be a pale shadow of Office's outline functionality, which allows selective expanding and collapsing of text—the main benefit of outlining.

Calculating Numbers

OpenOffice.org's answer to Microsoft Excel is Calc. It has most if not all of the features offered by Excel, including macros, which are organized into nine groups: Depot, Euro, FormWizard, Gimmicks, ImportWizard, Schedule, ScriptBindingLibrary, Template, and Tools.

To start a new spreadsheet in Calc, simply click the Spreadsheet button on the welcome screen.

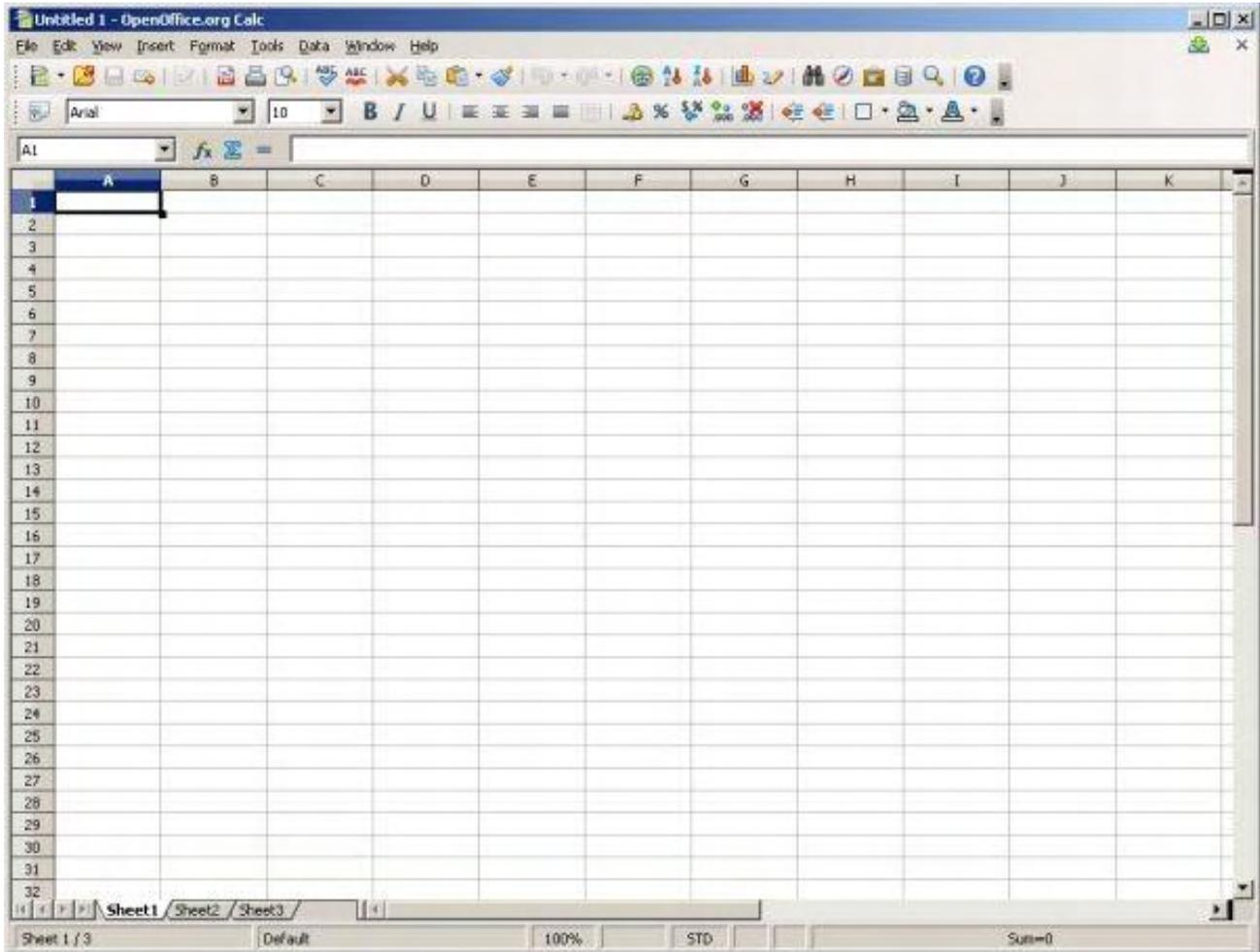


Figure 6. Calc new spreadsheet.

Like Writer, Calc has built-in formatting and styles, as well as the ability to embed images and other objects.

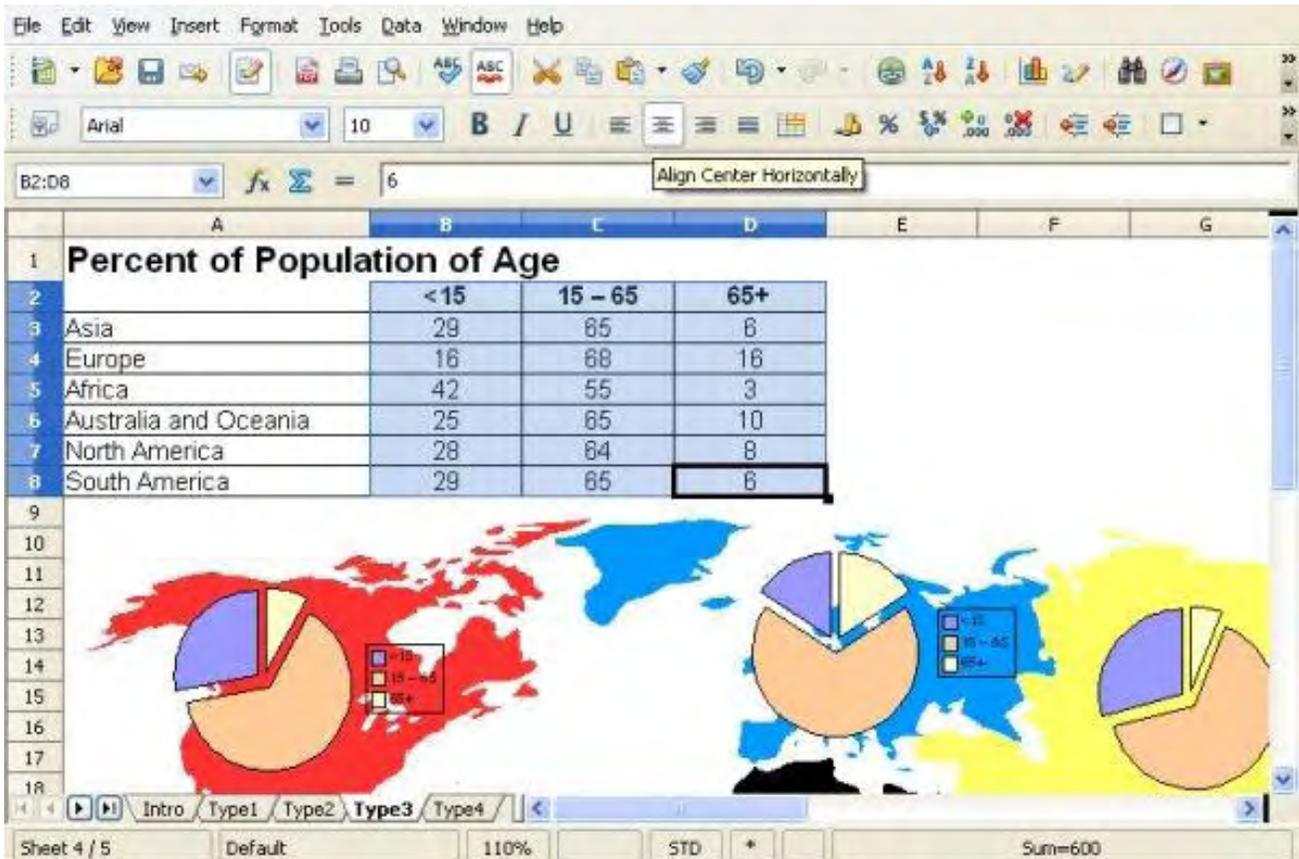


Figure 7. Calc existing spreadsheet.

Working on spreadsheets may be dry and dull, but Calc does have a playful side, in the form of a computer "Easter egg": In any cell, type in =Game("StarWars"), to start up a small game similar to Space Invaders, the classic video game.



Figure 8. Calc Star-Wars game.

The only downside to the game is that the response time of the rocket firings is quite slow, or so I've been told.

Impress Your Boss

Even though documents and spreadsheets are the most common types of documents created using office productivity suites, there are times when nothing can take the place of an eye-catching presentation, an attractive drawing, a well-structured database or, on that rare occasion, a math formula. OpenOffice.org can answer all of these needs.

To put together a multimedia presentation, consider using Impress, which supports numerous fonts, clip art (two-dimensional and three-dimensional), animation, drawings and special effects. You can create a new presentation from scratch, or by using a template wizard that steps you through the process of selecting such options as background colors and patterns, presentation title, image display speed, and more. When you are all finished, your complete presentation can be saved as a PDF file, an HTML page, or even as a Flash movie.

If you need to develop a drawing, you don't have to resort to Microsoft Paint, the applet built into Windows. OpenOffice.org has Draw, which makes it possible to create everything from simple drawings to sophisticated diagrams and graphics. It allows you to manipulate objects, rotate them in two- and three-dimensional space, and create photorealistic images utilizing sophisticated rendering. Furthermore, if the drawing that you are creating is some sort of diagram—such as a company org chart or software flowchart—you can use Draw's smart connectors to link together all of the constituent components. In most cases, you can avoid having to download and install a dedicated diagramming tool.

Creating a desktop database is usually a real chore—if not a daunting challenge—and Base makes it much easier to create and later modify tables, queries, forms, and reports based on that data. The default database engine is Base's own built-in HSQL, or you can use a different one, though it is not obvious as to how to specify which engine to use, in the

database-creation wizard. You can view your data using design views, wizards, or SQL, for those people who understand how to write such queries.

While the need to create a mathematical formula is typically limited to the academic world, if such a need arises, you can try using the Math program built into OpenOffice.org. It supports the importation of externally created formulas that have been saved in one of several formats—ODF, OpenOffice.org 1.0, StarMath, and MathML (www.w3.org/Math/).

Google Docs was mentioned earlier, and anyone who has documents stored in that system, or would like to export documents from OpenOffice.org and move them up into "the cloud," be sure to check out the Download Squad article (www.downloadsquad.com/2007/12/14/save-your-openoffice-org-docs-to-google-docs-and-vice-versa/) that explains how to transfer documents back and forth between the two systems.

This information should be enough to get anyone started in trying out OpenOffice.org, a promising alternative to Microsoft Office.

Michael J. Ross is a Web developer (www.ross.ws), writer, and freelance editor. He creates Web sites that help entrepreneurs turn their ideas into profitable online businesses.

[Return to Table of Contents](#)

OpenOffice.org Writer Macros

“Take the tedium out of repetitive editing tasks.” by Michael J. Ross

OpenOffice.org, the free and powerful office productivity suite that is gaining fans worldwide, supports the creation and use of macros, which are the best way to take the tedium out of repetitive editing tasks.

If you make heavy use of a word processor, spreadsheet application or text editor, you may find yourself performing the same editing operations repeatedly. For instance, if you are working at a company that is undergoing a name change, you might be responsible for updating the company name in dozens if not hundreds of documents. If all of these documents were plain text—such as HTML files on the company's Web site—then you could update all of them in one fell swoop (usually referred to as "global search and replace") using any powerful programmer's editor.

But the documents in question are much more likely to have been created using a word processing or spreadsheet program, and thus will not be in plain text. Consequently, the single global-search-and-replace method is not going to do the trick, because the text-replace feature in the typical office productivity application can affect only the document you are currently editing. In fact, if you are making changes to a spreadsheet workbook that contains multiple worksheets, for example, then any search-and-replace action performed using the Edit menu will update only a single one of those worksheets. You would have to go to each one of the worksheets and repeat the same action for each.

The same predicament exists in our name-change example, because when you begin the task of updating all of those company documents, you would have to open up every one of them individually, click on the Edit menu, and choose whatever menu item is provided for replacing text. If you are lucky, the string of text to search for, and the string with which to replace it, would already be populated in the dialog box's fields, or at least available in drop-down lists. But even then, the process would be quite tedious. Wouldn't it be great if you could save that search-and-replace action somehow, and execute it with a single keystroke?

That is what macros are all about. Think of a macro as a series of actions grouped together—like a miniature computer program that can be called upon at any time, to perform some useful task inside of your application. A macro is always given a name, so as to be easily referenced at any time in the future. Macros are either stored inside of a special file that is available to the parent application (word processing and spreadsheet programs invariably are designed this way), or they are stored in separate text files (this approach is largely limited to programmer's editors).

OpenOffice.org (www.openoffice.org/), the free and powerful office productivity suite that is gaining fans worldwide, supports the creation and use of macros to the same extent as does its main rival, Microsoft Office. Writer is the word processing component of OpenOffice.org; its Office counterpart is Word. Both applications support macros, using quite different variations of the BASIC programming language.

There is an extensive amount of information that you would need to learn in order to master the development of macros in Writer, so in this article we can only scratch the surface. But it will be enough to give anyone an idea as to what is involved and how to begin creating your own macros.

The Easy Way

If all of the actions that you wish to capture inside a macro can be performed entirely using the keyboard and pointing device, then the fastest way to make a new macro is to use the macro-recording feature built into the given application. Let's use our earlier example to illustrate how to do this in OpenOffice.org's Writer. Specifically, we will replace all occurrences of the string "Acme, Inc." with "XYZ, Inc." in the current document.

Begin the recording process by choosing the menu item Tools/Macros/Record Macro. A small dialog box will pop up, containing a single Stop Reporting button.



Figure 1. Dialog to stop recording.

Choose the menu item Edit/Find & Replace, type "Acme, Inc." in the Search for field and XYZ, Inc. in the Replace with field, and then click the Replace All button. If any instances of the search string were visible on the screen at the time that you executed the replacement, they should all have been changed to the replacement string. After acknowledging that the replacements have been done, click the Close button on the Find & Replace dialog box, and then click the Stop Recording button mentioned earlier.

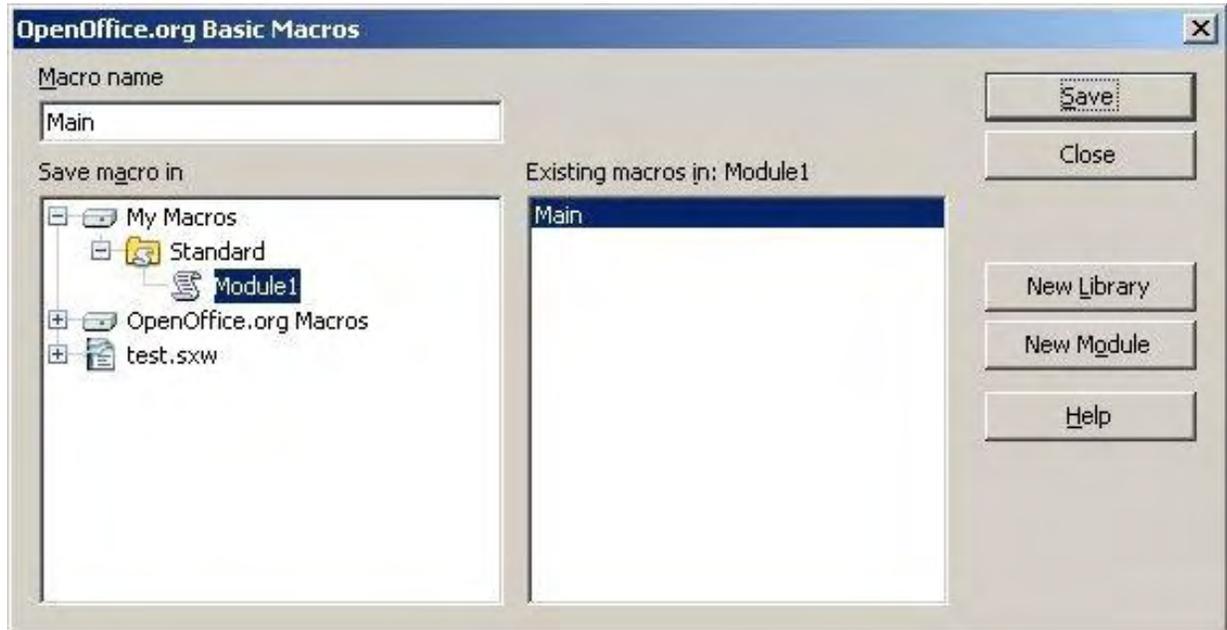


Figure 2. Dialog to save macro.

You will be prompted to save the macro that you just created. Let's give it the name "find_and_replace_company_name" and save it in the default module, Module1. To test the macro, undo the text replacement that you just performed (Edit/Undo); all instances of the old company name should be restored. Then choose the menu item Tools/Macros/Run Macro. In the dialog box that pops up, locate the macro you created.

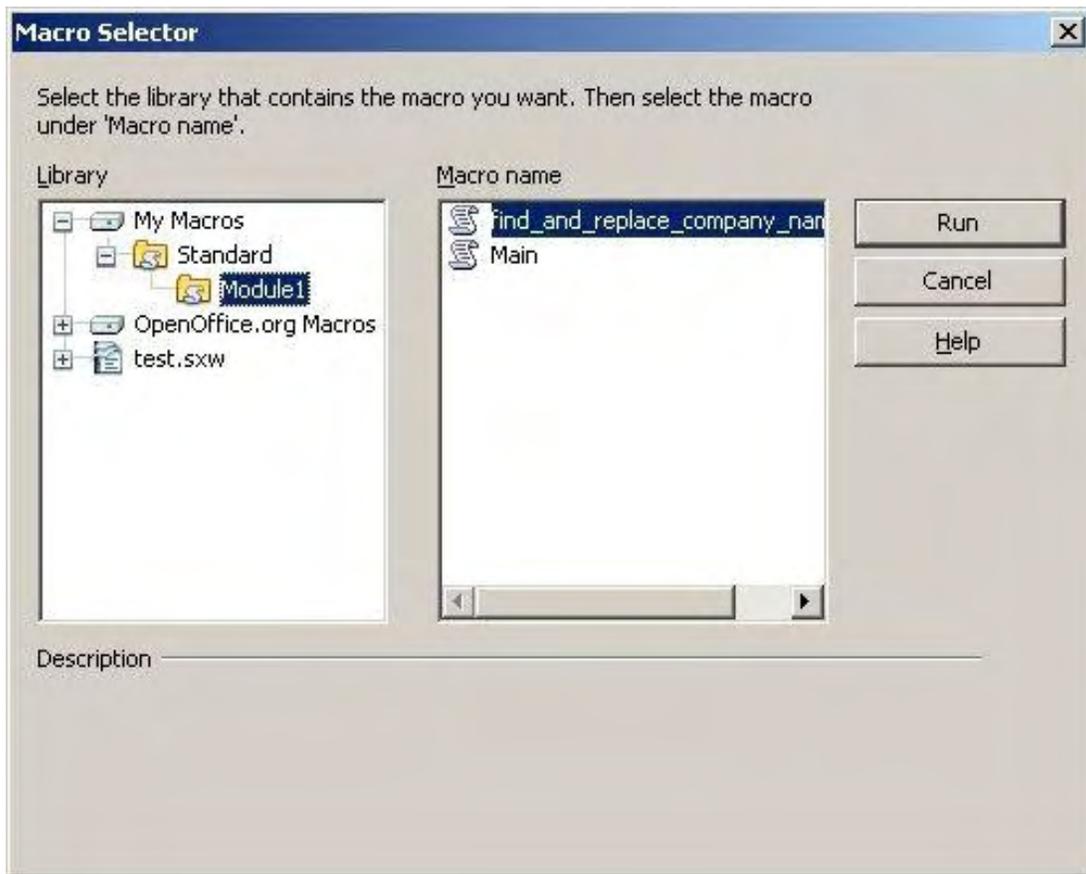


Figure 3. Dialog to run macro.

Click the Run button, and you should see the earlier text replacements performed again. You can then assign that macro to a new menu item, using the menu item Tools/Macros/Organize Macros/OpenOffice.org Basic/Assign.

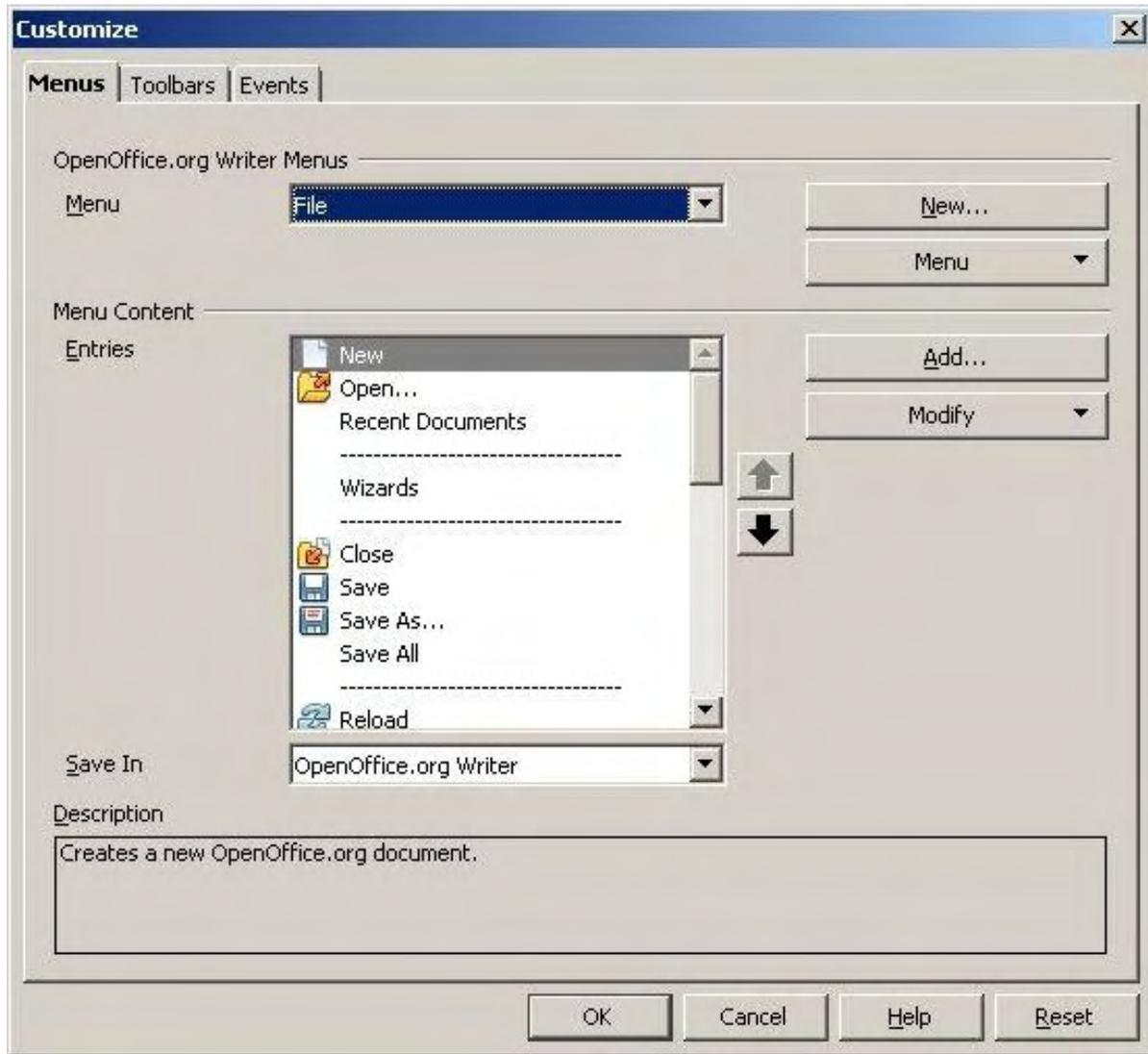


Figure 4. Dialog to assign macro.

At this point you may be wondering what the instructions (i.e., the source code) inside the macro look like. Choose the menu item Tools/Macros/Organize Macros/OpenOffice.org Basic. (We are using the first item on that cascading menu, because recorded macros are coded using the OpenOffice.org BASIC language.)

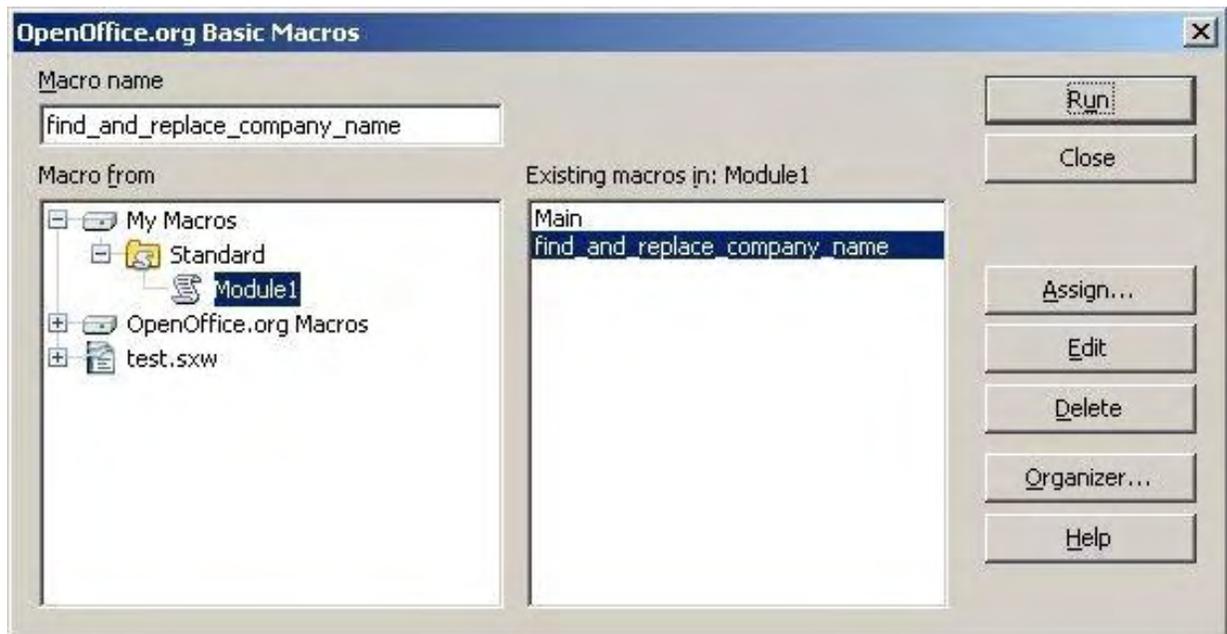


Figure 5. Dialog to organize macros.

By default, the most recently created macro is highlighted already. Click the Edit button. In the window that pops up, you will see the code for `find_and_replace_company_name`, as well as any other macros you may have created. But the edit cursor will be positioned at the line `sub find_and_replace_company_name`. The term "sub" is short for "subroutine," since that is the name used for any function in OpenOffice.org BASIC. The full code listing is shown below, except for the comments (any line beginning with "rem" — short for "remark").

```

sub find_and_replace_company_name
dim document as object
dim dispatcher as object
document = ThisComponent.CurrentController.Frame
dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")
dim args1(18) as new com.sun.star.beans.PropertyValue
args1(0).Name = "SearchItem.StyleFamily"
args1(0).Value = 2
args1(1).Name = "SearchItem.CellType"
args1(1).Value = 0
args1(2).Name = "SearchItem.RowDirection"
args1(2).Value = true
args1(3).Name = "SearchItem.AllTables"
args1(3).Value = false
args1(4).Name = "SearchItem.Backward"
args1(4).Value = false
args1(5).Name = "SearchItem.Pattern"
args1(5).Value = false
args1(6).Name = "SearchItem.Content"
args1(6).Value = false
args1(7).Name = "SearchItem.AsianOptions"
args1(7).Value = false
args1(8).Name = "SearchItem.AlgorithmType"
args1(8).Value = 0
args1(9).Name = "SearchItem.SearchFlags"
args1(9).Value = 65536
args1(10).Name = "SearchItem.SearchString"

```

```

args1(10).Value = "Acme, Inc."
args1(11).Name = "SearchItem.ReplaceString"
args1(11).Value = "XYZ, Inc."
args1(12).Name = "SearchItem.Locale"
args1(12).Value = 255
args1(13).Name = "SearchItem.ChangedChars"
args1(13).Value = 2
args1(14).Name = "SearchItem.DeletedChars"
args1(14).Value = 2
args1(15).Name = "SearchItem.InsertedChars"
args1(15).Value = 2
args1(16).Name = "SearchItem.TransliterateFlags"
args1(16).Value = 1280
args1(17).Name = "SearchItem.Command"
args1(17).Value = 3
args1(18).Name = "Quiet"
args1(18).Value = true
dispatcher.executeDispatch(document, ".uno:ExecuteSearch", "", 0, args1())
end sub

```

How does this code compare to that of an identical macro recorded in Microsoft Word (version 97), for instance?

```

Sub find_and_replace_company_name()
  Selection.Find.ClearFormatting
  Selection.Find.Replacement.ClearFormatting
  With Selection.Find
    .text = "Acme, Inc."
    .Replacement.text = "XYZ, Inc."
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
  End With
  Selection.Find.Execute Replace:=wdReplaceAll
  Selection.Find.Execute
End Sub

```

The Word code is recorded in the macro language VBA (Visual Basic for Applications). While it is certainly more concise than the OpenOffice.org code, in some respects VBA can be just as inelegant and difficult to write from scratch (which we will address next).

The Not-So-Easy Way

Space limitations do not allow us to get into the details of how to write OpenOffice.org macros from whole cloth. In fact, an entire book could be written on the subject. But we will consider some resources, for any reader who would like to pursue the subject further.

Perhaps the best place to start is the OpenOffice.org BASIC Programming Guide (wiki.services.openoffice.org/wiki/

Documentation/BASIC_Guide), which is organized as a wiki. It can also be downloaded as a PDF or ODT file, in case you would like to print it or do some work offline and still be able to reference the information. The programming guide explains the OpenOffice.org BASIC language, the run-time library, and the Uno API (Universal Network Objects application programming interface). The guide includes sections on how to write macros to work with Writer documents, as well as documents generated by the other components of OpenOffice.org—Calc, Draw, Impress, and Base.

While the sample code listed above may look daunting to some at first glance, developing macros in OpenOffice.org's flavor of BASIC is not too difficult to learn and, as with any programming language, gets easier with experience.

The Non-Basic Way

All of the macro development discussed so far utilizes some variation of BASIC. But unlike Microsoft Word, Writer is not limited to BASIC. Code for Writer functionality can also be written in a number of interpretive languages: JavaScript (en.wikipedia.org/wiki/JavaScript) (a language typically employed for client-side Web programming, since it is designed to run in a Web browser), BeanShell (www.beanshell.org/) (a subset of Java), and Python (www.python.org/) (a powerful object-oriented language gaining in popularity).

For instance, if you wanted to create a JavaScript macro for Writer, from scratch, you would select the Tools/Macros/Organize Macros/JavaScript menu item, create a new library, and then create a new macro within that library. If you were to open up that new macro using the Edit button, you would see that Writer has already provided some sample code that inserts the message "Hello World (in JavaScript)" at the end of the current document. The code is listed below, with all comments removed:

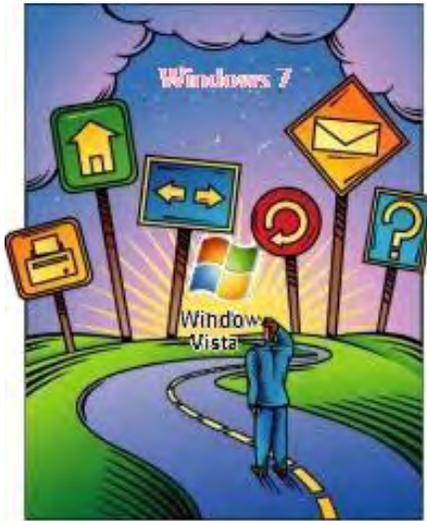
```
importClass(Packages.com.sun.star.uno.UnoRuntime);
importClass(Packages.com.sun.star.text.XTextDocument);
importClass(Packages.com.sun.star.text.XText);
importClass(Packages.com.sun.star.text.XTextRange);
importClass(Packages.com.sun.star.frame.XModel);
oDoc = UnoRuntime.queryInterface(XModel,XSCRIPTCONTEXT.getInvocationContext());
if (!oDoc)
    oDoc = XSCRIPTCONTEXT.getDocument();
xTextDoc = UnoRuntime.queryInterface(XTextDocument,oDoc);
xText = xTextDoc.getText();
xTextRange = xText.getEnd();
xTextRange.setString("Hello World (in JavaScript)");
```

The first section of code makes available five API classes, and the code that follows creates some needed objects, and ends by writing out the desired message string at the end of the text range within the current document. Creating new macros in BeanShell and Python results in code that performs a similar operation.

As should be evident, there is so much that you could learn about writing macros for Writer, and there is even a choice of computer language—a rare bonus among applications that support macros. If you choose to try writing some macros of your own, not only is the aforementioned programming guide available, but be sure to check out the OpenOffice.org Community Forum (user.services.openoffice.org/en/forum/). You may soon discover how macros are the best way to take the tedium out of repetitive editing tasks.

Michael J. Ross is a Web developer (www.ross.ws), writer, and freelance editor. He creates Web sites that help entrepreneurs turn their ideas into profitable online businesses.

[Return to Table of Contents](#)



Windows Vista Tips and Tricks

(and some Windows 7)

[Windows Vista and Windows 7 Tips and Tricks](#)
“Windows 7 Upgrade Deal Ends July 11! Does it really matter?” by Jack Dunning

Microsoft is anxious for you to pre-order Windows 7 upgrades. Should you rush out and purchase it? Also, more on problems with the Internet Explorer 8 update, and a tip on easier file-saving in Windows 7.

Windows 7 Upgrade Deals

Microsoft is anxious for you to pre-order Windows 7 upgrades. (The offer is good only until tomorrow, July 11, 2009. Will the company extend it?) Microsoft is offering a discount on the Home Premium and Professional versions (\$49.99 and \$99.99 respectively; see Figure 1). No discounts for the Ultimate version. I'm thinking that Microsoft doesn't want to push the Ultimate version because it doesn't offer that much more than the Professional.

Which Windows 7 is best for you?		 58% off Windows 7 Home Premium Upgrade Upgrade Buy \$149.99 \$49.99	 50% off Windows 7 Professional Upgrade Upgrade Buy \$199.99 \$99.99	 Windows 7 Ultimate Upgrade Upgrade Buy \$219.99 \$219.99
View Windows 7 Full Versions				
Makes the things you do every day easier with improved desktop navigation	●	●	●	●
Faster and easier to launch applications and find the documents you use most often.	●	●	●	●
Make your web experience faster, easier, and safer than ever with Internet Explorer 8.	●	●	●	●
Easy to create a home network and connect your PCs to a printer with HomeGroup.	●	●	●	●
Watch many of your favorite TV shows for free when and where you want with Internet TV. 1	●	●	●	●
Run many existing Windows XP productivity applications in Windows XP Mode. 2		●	●	●
Connect to company networks easily and more securely with Domain Join.		●	●	●
Recover your data easily with automatic backup to your home and business network.		●	●	●
Help protect your data on your PC and portable storage devices against loss or theft with BitLocker.			●	●



Figure 1. Microsoft pre-order of Windows 7 upgrades.

Some people are suggesting that you buy two copies of the upgrade now and sell one copy at just below full price when you receive delivery. Then, effectively you will get the upgrade for free. The price looks good, but the free upgrade strategy may account for a good portion of the brisk sales. Remember, if you buy a computer with Vista between now and the release date, you qualify for a free Windows 7 upgrade.

If you're using Vista and are happy with it (yes, there are happy Vista users), then you may not be in a rush to upgrade—if ever. There will be yet-unknown problems associated with upgrading from Vista to Windows 7. I don't know what they are, or whether they will be Vista or 7 problems, but there will be something. I expect that many of these discounted Windows 7 upgrades will go unused, as many XP PCs will probably be just too old to take it (underpowered—there are still minimum requirements).

Pricing for the full Windows 7 package is about the same as it is for Vista—slightly cheaper for the Home Premium version. The best way for most people to get it will be with a new computer. Generally, we don't think about the price of the operating system when we buy a new computer—unless we're building it ourselves. Most of us have old copies of XP lying around. We can always install that XP first, then use a cheaper upgrade version of Windows 7.

You can go the Microsoft Upgrade page (store.microsoft.com/microsoft/Windows-Windows-7/category/102?WT.mc_id=msccomhpfeature_win7) and see for yourself.

Internet Explorer 8 Update

I've been working with the 64-bit version of Internet Explorer 8. While it loads and runs fast, I've had just as many problems with it crashing as I had with the 32-bit version—worse than IE7. Plus, IE8 is not supported by Adobe Flash and most other third party add-ons. Although IE8 has a number of cool features, operationally it feels like a step backward. I might just move to Google Chrome as my primary working browser.

I have been using Google Chrome on one of my Vista machines and have had no problem with it crashing. As for the InPrivacy mode in IE8 touted in the latest Web commercial (properly slammed by Wally this week), Chrome has an Incognito mode that does the same thing. I would use Firefox, except I haven't had time to debug the AJAX problem in version 3.X. My AJAX programming works in every other browser including Firefox 2.X. What happened?

Easier File Saving

When using Windows Explorer in Vista, the navigation pane has the added feature of Favorites. These are shortcuts that you can add to navigation for those folders that you use most often. In Windows Vista, the Microsoft Save window has the same look as the navigation pane in Windows Explorer—at least for those programs that support it. OpenOffice.org is one of those programs, as seen in Figure 2. I find this very convenient when working with multiple files and programs.

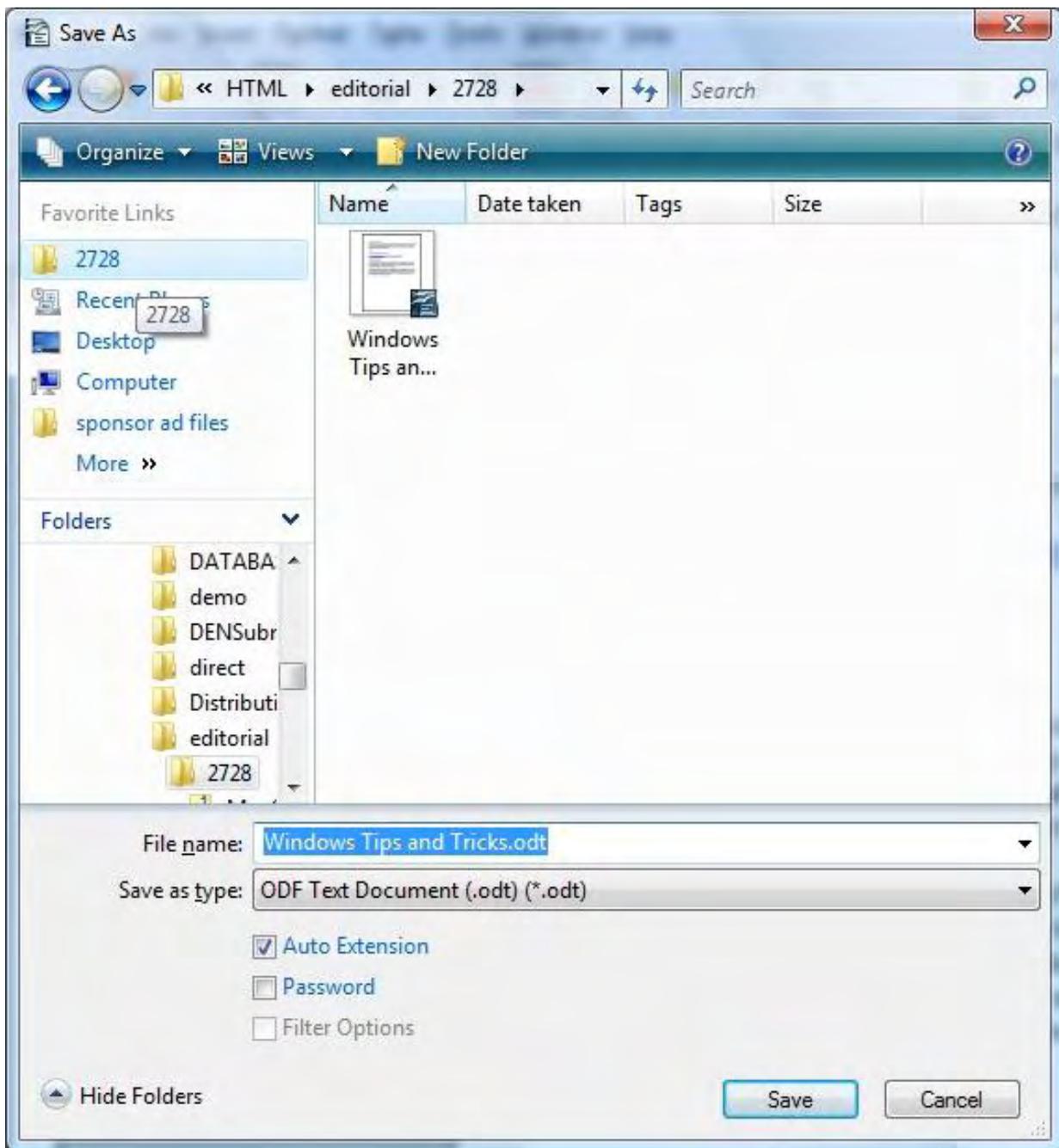


Figure 2. Windows Vista "Save As" window in OpenOffice.org.

However, many programs do not support this format. When using Adobe Acrobat 8, the Save As window has options that are almost as convenient. In most cases, I can select Recent Places to get to the folders I need quickly (see Figure 3). Since I'm growing accustomed to using the Favorites in the example above, I would prefer that the programs conform to that format, although picking from Recent Places is almost as good.

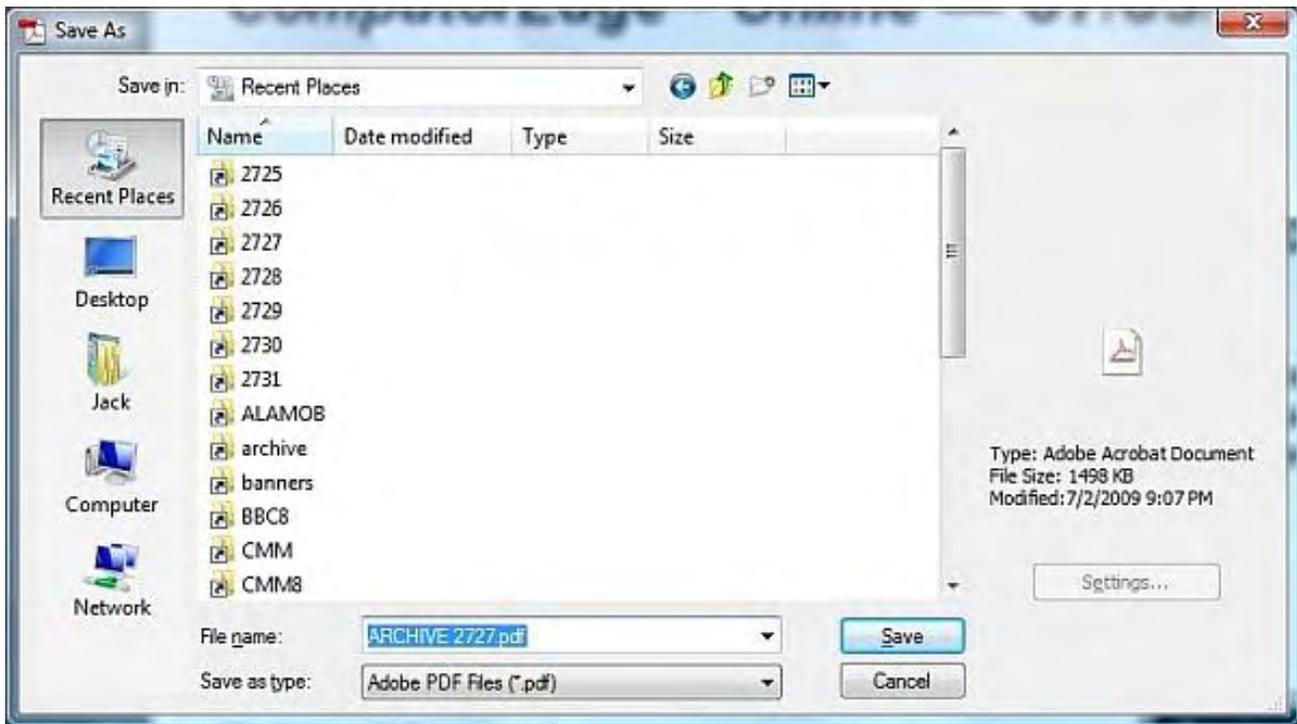


Figure 3. Windows Vista "Save As" window in Adobe Acrobat 8.

At this point, the software I find the most annoying is that which forces me to dig through the folder hierarchy to find the proper location for saving, although that is usually restricted to older utility programs. I guess I'm getting lazy.

Windows 7 is adding Libraries to the list of navigation items (see Figure 4). In both Windows Explorer and the various Open and Save As windows, you will tailor your navigation to suit your needs. In the long term, this is a good thing. The amount of data we are storing on our hard drive is too big and varied to organize only with a tree structure.



Figure 4. Windows 7 "Save As" window for the Paint program.

Jack is the publisher of *ComputerEdge* Magazine. He's been with the magazine since first issue on May 16, 1983. Back then, it was called *The Byte Buyer*. His Web site is www.computoredge.com. He can be reached at ceeditor@computoredge.com

[Return to Table of Contents](#)



Wally Wang's Apple Farm

“The Best Software Is Often Free” by Wally Wang

In case you'd rather not pay the upgrade fee for the iLife suite, or simply find that iLife doesn't meet your needs, here are some alternative programs that are completely free. Also, a look at Microsoft's scatterbrained marketing campaign, and a tip on using Dashboard Web Clips.

Wally Wang's Apple Farm

One of the best "free" collections of programs that you get with every Macintosh is iLife, which includes iPhoto (for editing and organizing digital photos), iMovie (for editing digital video), iWeb (for creating Web pages) and GarageBand (for editing podcasts or creating your own music).

Every Mac user should take the time to explore these iLife programs, although it's unlikely that everyone will need or use every iLife program. Anyone with a digital camera or camcorder will likely appreciate iPhoto and iMovie. On the other hand, iWeb is more suitable for novices who want to create simple Web pages, but nothing too sophisticated, while GarageBand is probably the most specialized program of the suite.

The latest iLife suite costs \$79, so if you have an old Macintosh, you'll need to pay for the privilege of upgrading your old copy of iLife. In case you'd rather not pay this upgrade fee, or simply find that iLife doesn't meet your needs, here are some alternative programs that are completely free.

If you're intrigued by GarageBand but find it a bit too intimidating, try Audacity (audacity.sourceforge.net). GarageBand is still superior if you want to make your own music or learn to play an instrument, but if you just need to record and edit audio, Audacity is far simpler and straightforward.

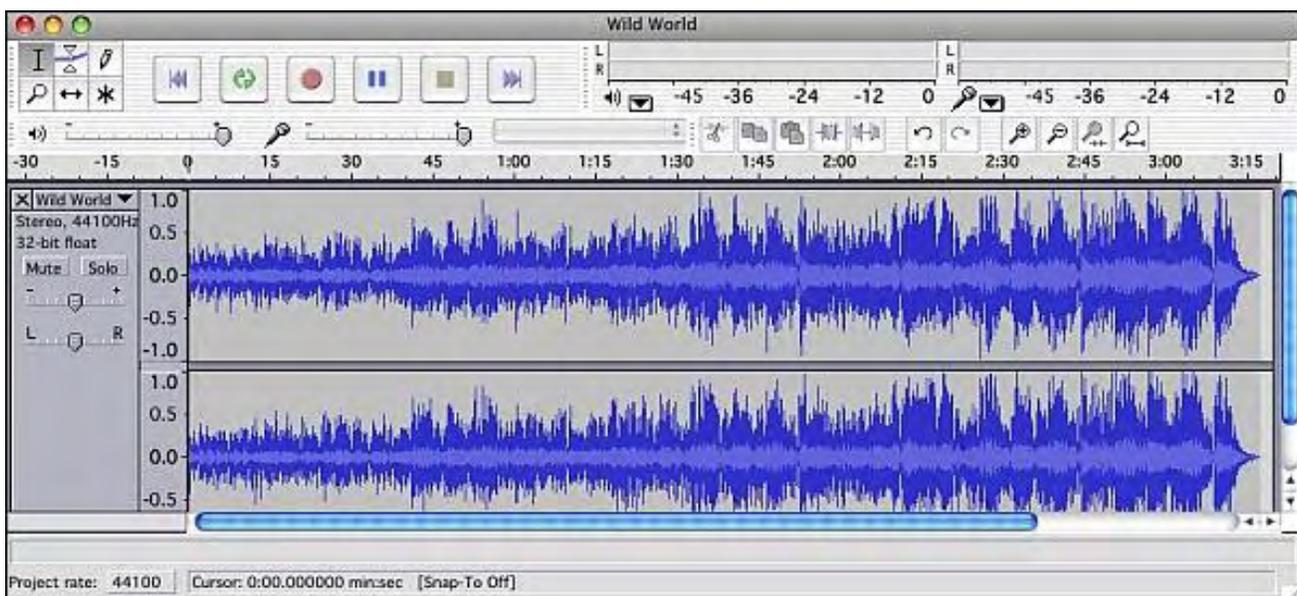


Figure 1. Audacity lets you record and edit audio.

Designing Web pages isn't difficult, but it does take time to learn to do it well. Since iWeb is geared for novices, it makes creating Web pages easy, but shields you from the actual HTML code needed to create unique effects on a Web page. In case you find iWeb a bit too limiting, try Nvu (net2.com/nvu).

Nvu lets you write and edit the raw HTML code of any Web page, which makes it more powerful for intermediate and advanced Web designers, but a bit intimidating for complete novices who just want to slap a Web page together in a few seconds.

Think of Nvu as a poor man's Dreamweaver. It's not quite as sophisticated or powerful as Dreamweaver, but if you're itching to create and edit HTML code, Nvu can help you get started.

MacWrite, MacPaint, and MacDraw Alternatives

In the old days, many computers used to come with bundled software that usually included a word processor and other standard programs to help you get started using your computer right away. The first Macintosh computers came bundled with MacWrite (a word processor) and MacPaint (a painting program), but you can mimic this bundled software starting with Bean (www.bean-osx.com/Bean.html) and Paintbrush (paintbrush.sourceforge.net).

Although Bean is a simple word processor, it's more sophisticated than the free TextEdit program that comes with today's Macs. Paintbrush is a full-featured paint program that lets you edit pictures or create your own.

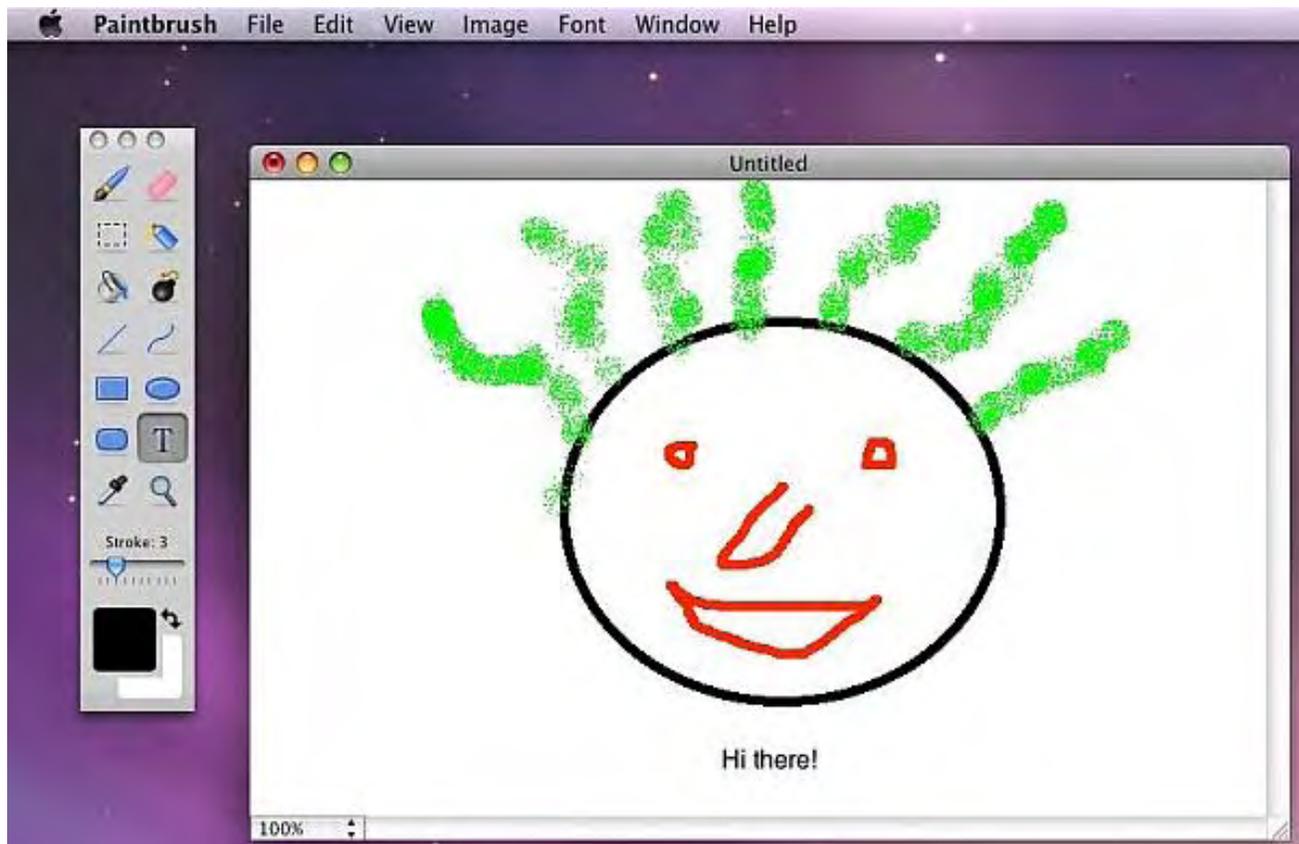


Figure 2. Paintbrush lets you edit or draw pictures.

If you need a drawing (vector-based) program similar to the old MacDraw, then grab a copy of DrawBerry (raphaelbost.free.fr/DrawBerry.html). While Paintbrush acts like a simple version of Photoshop, DrawBerry acts like a simple version of Adobe Illustrator.

Office Suite Alternatives

For those who need heavy-duty office programs, two popular choices include:

- OpenOffice.org (www.openoffice.org)
- Lotus Symphony (symphony.lotus.com/software/lotus/symphony/home.nsf/home)

Both programs are completely free and offer features similar to Microsoft Office along with the ability to open and edit Microsoft Office files. Although their names differ, Lotus Symphony is actually IBM's modified version of OpenOffice.org, so you essentially get the exact same features in both office suites.

The main reason you might want OpenOffice.org is that it includes a word processor, spreadsheet, presentation program, drawing program and database. The main reason to get Lotus Symphony is that it gives you only a word processor, spreadsheet and presentation program, but it offers a neater tabbed interface that lets you switch between the various components of the office suite as easily as clicking on a tab. If you've ever used tabs while browsing Web pages, you already know how to use the tabbed interface of Lotus Symphony.

Although Lotus Symphony lacks the drawing and database program of OpenOffice.org, you might not need those programs, so you probably don't want them cluttering up your computer. If you do need those programs, then you'll have no choice but to use OpenOffice.org.

Since both OpenOffice.org and Lotus Symphony can create and read Microsoft Office files, you can use these free programs instead of buying Microsoft Office for the Mac. The main advantage of Microsoft Office is that it comes loaded with dozens of templates that you can use and modify. With OpenOffice.org and Lotus Symphony, you have to create most documents from scratch.

With so many free programs available for the Macintosh, you should have no trouble finding the type of program you need. By using free software, you can save your money to spend on more important things in life, like food and gas.

More Puzzling Advertisements From Microsoft

After blowing \$10 million to hire Jerry Seinfeld for two nonsensical ads, Microsoft has hit a new low in its marketing campaign. It was bad enough that its "Laptop Marketing" series of ads deliberately deceived the public by showing a 13-inch MacBook, but displayed the higher price tag of a 17-inch MacBook Pro to suggest that the tiny 13-inch MacBook cost several hundred dollars more than it really did.

Now Microsoft has released a questionable online ad (www.youtube.com/watch?v=xB9fhjnJcB0) designed to promote its Internet Explorer 8 browser. In this ad, a wife asks her husband if she can use his laptop. He agrees, but after seeing the sites he had been visiting, the wife projectile vomits on the kitchen floor.

The husband then slips in her vomit and lands on his back while the woman continues to throw up over his legs and chest. Not content to just suggest she upchucks, this ad provides plenty of realistic-looking vomit on the floor, on the man, and dangling from the lower lip of the wife.

The first question that comes to mind is, "Who wrote this script and thought that realistic-looking vomit spewing out of a woman would help promote Internet Explorer 8 as a better browser than its rivals?" Second, "Who in the advertising agency approved this commercial and thought it would cast Microsoft in a favorable light?" Third, who in Microsoft saw this ad and thought, "Great! Projectile vomiting in gory detail is exactly the image we need to represent a Fortune 500 company!"

Advertisements are supposed to help sell products, but Microsoft seems to have missed that basic point in its advertising campaigns. What does showing Jerry Seinfeld and Bill Gates together in a shoe store and in an ordinary house have to do with Microsoft? And what does projectile vomiting have to do with promoting Internet Explorer 8? From what I can tell, absolutely nothing, and that's the huge problem with Microsoft's scatterbrained marketing campaign right there.

Whatever you think of Microsoft and its products, you have to wonder about any company that thinks projectile vomiting would be a great advertising tactic. When I first saw this projectile vomiting ad, I was absolutely convinced that an anti-Microsoft person created this as a gag to associate upchucking with Microsoft. Only later did I discover that

Microsoft itself created, paid for, and promoted this ad, presumably to appear "hip" and "humorous."

If being deliberately disgusting and resorting to juvenile humor is the best that Microsoft can do to promote its products, you have to wonder if anyone at Microsoft has any vision and direction for the company at all. After witnessing the debacle of the Seinfeld-Bill Gates ads, and now seeing this projectile vomiting ad, I'm more convinced than ever that Microsoft is simply going to shoot itself in the foot and be its own worst enemy.

Toss in confusing reports on upgrade prices for Vista users and rumors that Vista Ultimate users (who paid the most for "extra features" that Microsoft promised but never delivered) must now pay the most for an upgrade to Windows 7, you have to wonder, what is Microsoft thinking these days? If you thought General Motors imploded on itself through sheer incompetence, watching Microsoft's latest ads almost makes you feel sorry for the company as it finds more creative ways to embarrass itself in public—while wasting millions of dollars doing it.

After trying the disaster that was Vista, being confused by the Seinfeld-Bill Gates ads, and appalled by the latest Internet Explorer 8 vomit ad, I feel certain that Microsoft has lost its direction for good.

* * *

Safari on the Macintosh has a unique feature called Web Clips, which you can access by choosing File/Open in Dashboard. This command lets you highlight part of any Web page and make it appear as a Dashboard widget, which you can pop on the screen by pressing F12 at any time.

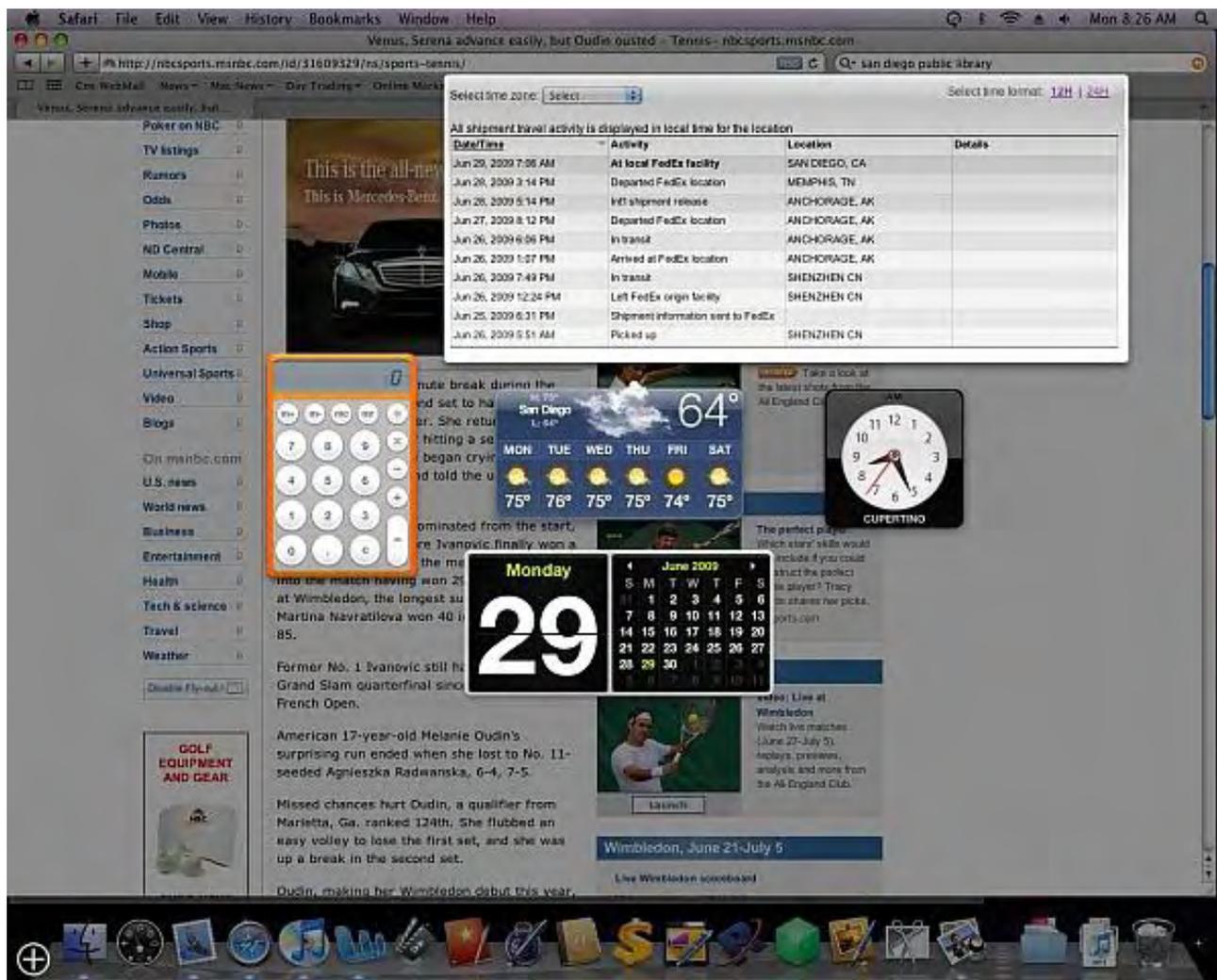


Figure 3. Tracking a FedEx shipment with a Dashboard Web Clip.

Recently, I purchased the latest iPhone 3GS through Apple's Store. Apple sent me a FedEx tracking number, which I

used to display the latest shipping results of my package. Using Safari's Web Clip feature, I copied this portion of the Web page as a Dashboard Web Clip.

Now, to track my package throughout the week, I didn't have to type in my FedEx tracking number by visiting the FedEx Web site over and over again. Instead, I just had to tap the F12 key to see the progress of my package in Dashboard. As my package got closer to San Diego, I could schedule my time to insure that I would be home on the day FedEx would deliver it to my door.

By helping you track changing information off a Web page, Safari and Dashboard can make your life easier. While not a revolutionary feature, it's just one more reason why I have more fun with my Macintosh than I ever did with any Windows PC that I ever owned.

In the early days, before Wally became an Internationally renowned comedian, computer book writer, and generally cool guy, Wally Wang used to hang around The Byte Buyer dangling participles with Jack Dunning and go to the gym to pump iron with Dan Gookin.

Wally is responsible for the following books:

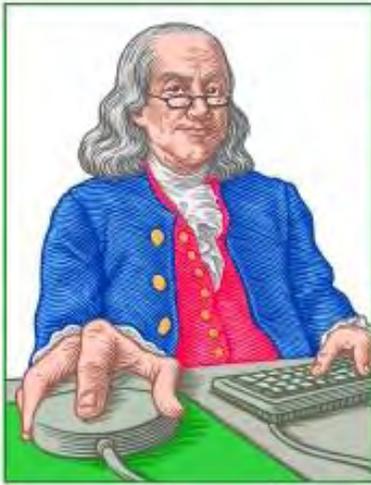
- Microsoft Office 2007 for Dummies (www.amazon.com/gp/product/0470009233?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0470009233)
- Beginning Programming for Dummies (www.amazon.com/gp/product/0470088702?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0470088702)
- Breaking Into Acting for Dummies with Larry Garrison (www.amazon.com/gp/product/0764554468?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0764554468)
- Beginning Programming All-in-One Reference for Dummies (www.amazon.com/gp/product/0470108541?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0470108541)
- Steal This Computer Book 4.0 (www.amazon.com/gp/product/1593271050?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1593271050)
- Visual Basic Express 2005: Now Playing (www.amazon.com/gp/product/1593270593?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1593270593)
- My New Mac (www.amazon.com/gp/product/1593271646?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1593271646)
- My New iPhone (www.amazon.com/gp/product/1593271956?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1593271956)
- Strategic Entrepreneurism with Jon Fisher and Gerald Fisher (www.amazon.com/gp/product/1590791894?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1590791894)

Every Saturday morning from 9:00 am - 10:00 am in San Diego, you can hear Wally with fellow co-hosts Dane Henderson and Candace Lee, on the radio show CyberSports Today (cybersportstoday.com/), which covers the video gaming industry on ESPN Radio 800 AM. Wally covers the military history side of the video game industry.

When not performing stand-up comedy or writing computer books, he likes to paper trade stocks with the video game Stock Reflex (www.plimus.com/jsp/download_trial.jsp?contractId=1722712&referrer=wwang), using the techniques he learned from a professional Wall Street day trader.

Wally can be reached at wally@computoredge.com and at his personal web site (www.wallacewang.com/).

[Return to Table of Contents](#)



LITTLE LINUX LESSONS

Little Linux Lessons:
Tips and Tricks from
Users

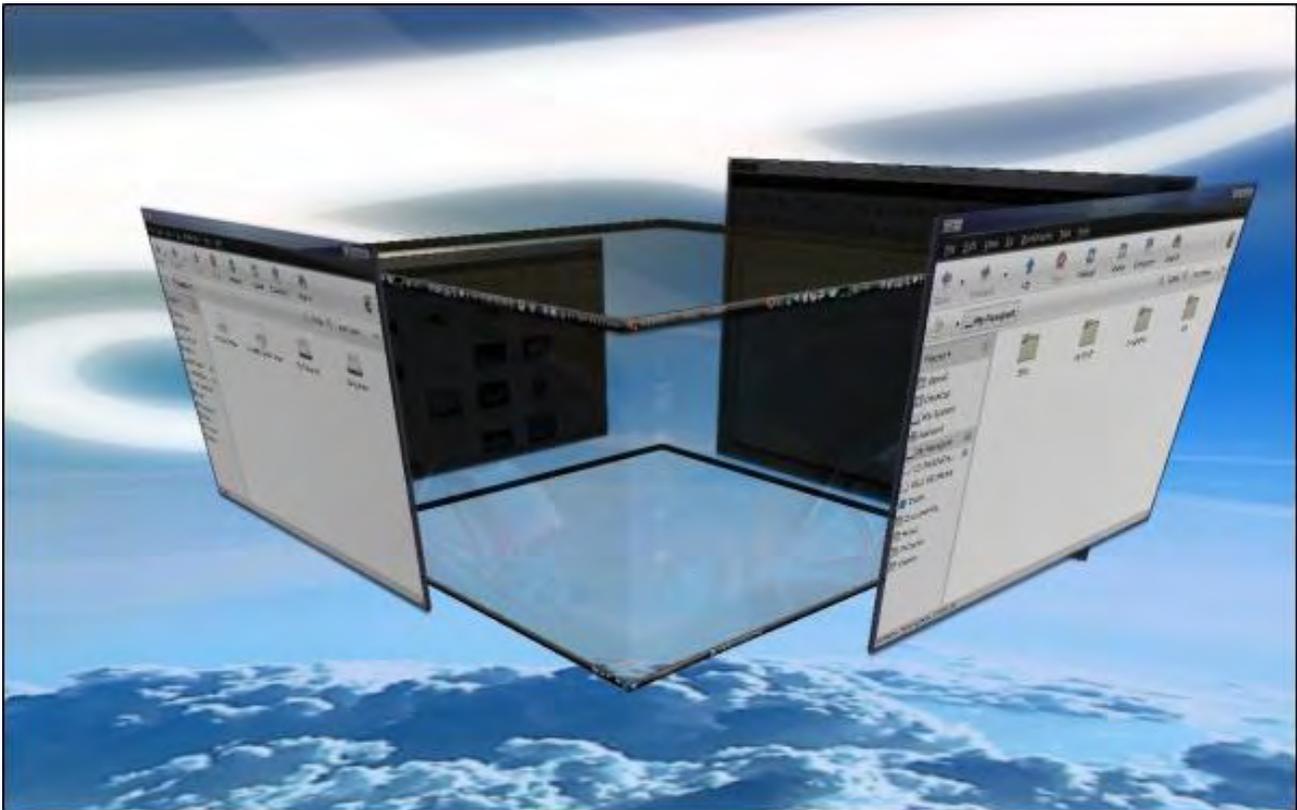
“Linux users share ideas and ask for help.” by ComputerEdge Staff

**“AN INVESTMENT
IN LINUX KNOWLEDGE
PAYS THE BEST
INTEREST.”**

A reader writes in with a tip on Compiz, a program that runs cool desktop effects in Ubuntu Linux.

Compiz Fusion on Ubuntu

One of the things that separate Ubuntu from Windows is its very cool desktop effects. What do I mean by cool? Here is a sample:

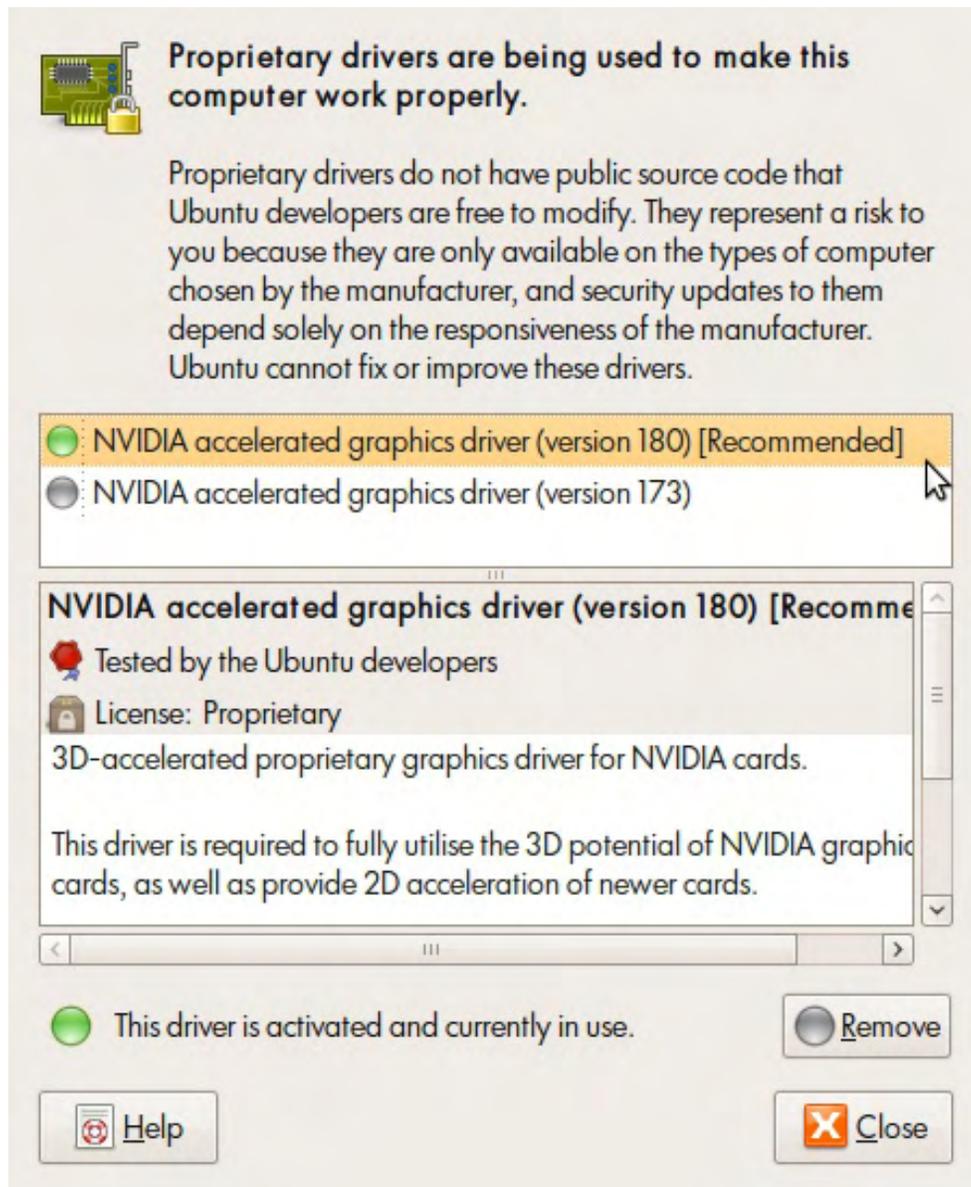


So, how do these effects get enabled? The program that runs these desktop effects is called Compiz Fusion (on older versions of Ubuntu, it is called just compiz). Step one, then, obviously, is to install Compiz if it is not already there. True, you can do that, but after installing it and following the directions afterward, you may find that it still does not work. Why? Because there is a step zero that comes before step one. Step zero is to install the drivers for your graphics card.

Step Zero: Installing the Drivers

This is easier than it sounds. With your cursor, go to System/Administration/Hardware Drivers. After Ubuntu searches

online for drivers, a window like this will pop up:



Select the recommended driver (if needed), and restart your computer. *Now*, you can move on to step one.

Step One: Installing Compiz Fusion

Go to System/Administration/Synaptic Package Manager, and click on the search button (Note: Do not just type in the search box, as the searches are not as thorough there). When the search window pops up, type in compiz. After the search has finished, select these packages:

compiz, compizconfig-settings-manager,
 compiz-fusion-plugins-main, compiz-fusion-plugins-extra,
 compiz-fusion-plugins-unsupported, emerald,
 fusion-icon, simple-ccsm

Some packages you may already have, so don't worry about those. Also, some packages depend on others to work, and a prompt will come asking if you want to install those too. Say yes. Afterward, click on Apply and go grab a sandwich while Ubuntu installs the packages.

Step Two: Finding Compiz

The settings for Compiz can be found in two places:

1. System/Preferences/CompizConfig Settings Manager
2. System/Preferences/Simple Compiz Config Settings Manager

The first manager can be very in-depth at times, because it allows you to control almost every aspect of your desktop effects. The second manager is more basic, but includes some tweaks that the other does not have, hence the reason both need to be installed.

Step Three: Push Buttons

Both these managers may seem a little overwhelming in the beginning (especially the first one). The best way to learn is to try out different effects. Some effects are very obvious, such as the desktop cube, while others are more subtle, such as wobbly windows (which make all your windows wiggle like Jell-O when you try to move them). Start some up and see how it works. If you don't like one, turn it off and try another. Almost everything with Compiz can be undone somehow, and unless you do something *really* drastic, you will not crash your machine completely.

Conclusion

Compiz is the best way to personalize your machine. With it, you can increase your functionality and, more importantly, wow all your Windows friends. While this is a general overview of installing Compiz, there is lots of info on specific plug-ins to be found online.

This tip was submitted by Dan Bartels of San Diego. Visit Dan's Blog (www.dansblog102.blogspot.com).

Give Us Your Linux Tips and/or Questions

If you have an opinion on these or other Linux topics, then please let us know. Also, if you have another Linux tip that works for you and would like to pass it along (or have a question), please drop us a line at Linux Lessons (ceeditor@computoredge.com).

This is a column for Linux and Unix-like operating system users. The goal is to give Linux users an opportunity to share tips, tricks and ideas with both fellow users and the *ComputerEdge* Linux newbies. Each week in this column, we will highlight the thoughts you submit to us. This is your column. As long as a submission is dealing with the Linux/Unix-like world, we want to share it.

The tips and tricks may be short or long, and can include graphics. If there is a little technique or program that you use on a regular basis, then we want to hear about it. You may also pose questions for other Linux users to answer. E-mail your ideas or questions to Linux Lessons (ceeditor@computoredge.com). Be sure to put the words "Linux Lessons" in the subject line so it won't get lost in junk mail. We depend upon you to make this column a success.

Jack Dunning

ComputerEdge

ComputerEdge always wants to hear from you, our readers. If you have specific comments about one of our articles, please click the "Tell us what you think about this article!" link at the top or bottom of the article/column. Your comments will be attached to the column and may appear at a later time in the "Editor's Letters" section.

If you want to submit a short "ComputerQuick Review", or yell at us, please e-mail us at ceeditor@computoredge.com.

[Return to Table of Contents](#)



Rob, The Computer Tutor

ComputerTutor Does Microsoft Access

“Troubleshooting: Name Parsing” by Rob Spahitz

Last week, we created a procedure to allow a user to enter a full name and let us split it into a first and last name to put it into the corresponding fields. It didn't quite work as expected, so we'll explore ways to help us learn how to deal with such issues.

Last week, we created a procedure to allow a user to enter a full name and let us split it into a first and last name to put it into the corresponding fields. It didn't quite work as expected, so we'll explore ways to help us learn how to deal with such issues.

As usual, if you'd like to see previous version of the database, look for them at www.dogopoly.com/ce.

As a refresher, Figure 1 shows the text box we added to the header and the corresponding button to split the name and put its pieces into the person subform.

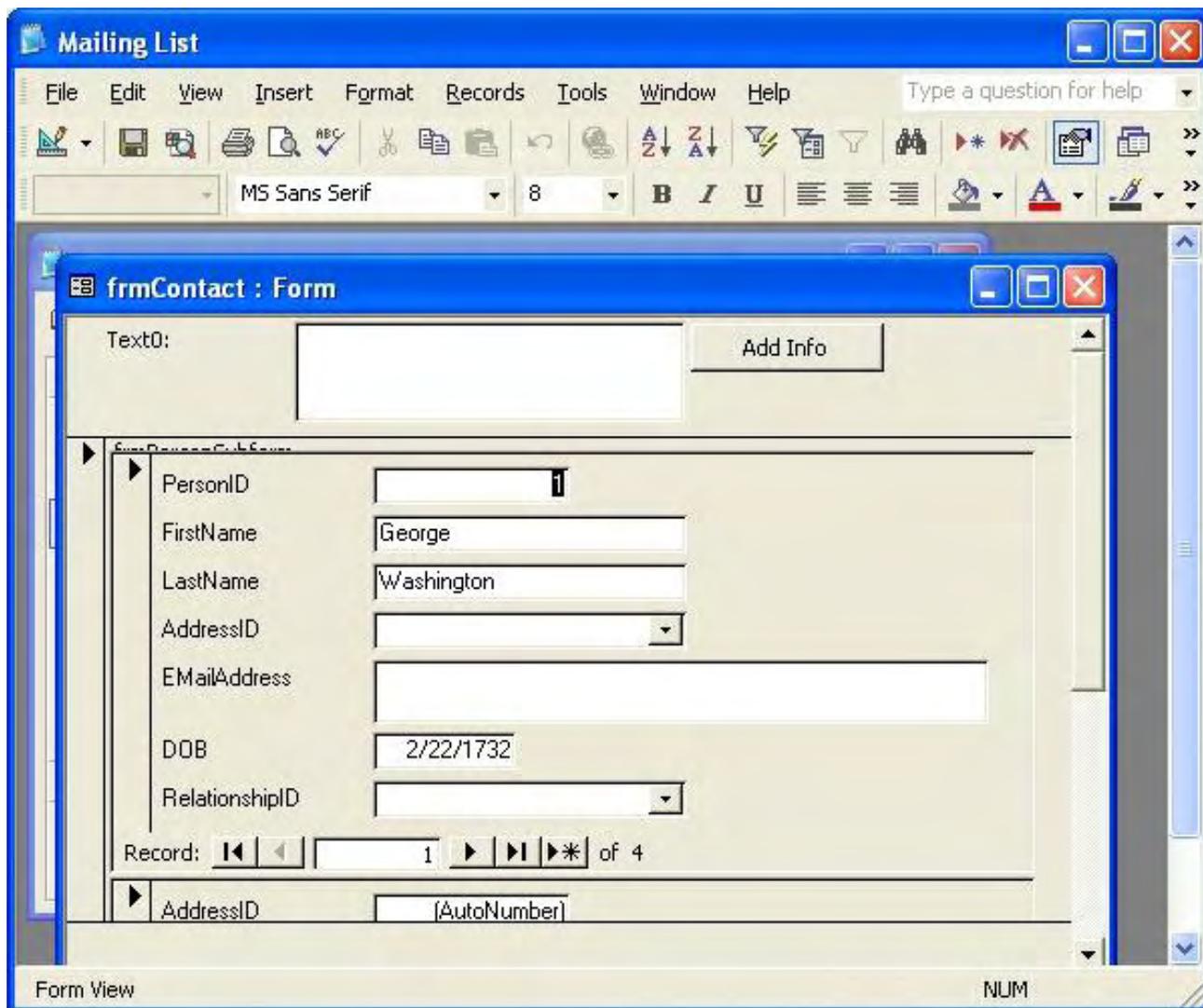


Figure 1. Input Box for Name.

When we tried our box with Abe Lincoln, it put Abe into the LastName field and didn't put Lincoln anywhere. Let's review the code and see if we can find a solution:

```
Private Sub cmdAddInfo_Click()
    AddPerson GetPersonName()
    'AddAddress GetAddressInfo()
End Sub
Private Function GetPersonName() As String
    Dim strPersonName As String

    strPersonName = Split(Me.txtContact.Value & vbNewLine, vbNewLine)(0)

    GetPersonName = strPersonName
End Function
Private Sub AddPerson(PersonName As String)
    Dim strPersonNames() As String
    Dim iNameCount As Integer
    Dim strFirstName As String
    Dim strLastName As String

    strPersonNames = Split(PersonName, " ")
    iNameCount = UBound(strPersonNames)

    Select Case iNameCount
        Case 0
            ' No names to add so skip it
            strFirstName = ""
            strLastName = ""
        Case 1
            ' Use as last name only
            strFirstName = ""
            strLastName = strPersonNames(0)
        Case 2
            strFirstName = strPersonNames(0)
            strLastName = strPersonNames(1)
        Case Else
            If InStr(strPersonNames(iNameCount), ".") > 0 Then
                ' contains a dot so use previous as last name
                strLastName = strPersonNames(2)
                ' check for . in first name
            Else
                strLastName = strPersonNames(3)
                ' try to find first name
            End If
            If InStr(strPersonNames(0), ".") > 0 Then
                ' dot found in first name so skip it
                strLastName = strPersonNames(1)
            Else
                ' use first word as first name
                strLastName = strPersonNames(0)
            End If
        End Select
    End Select
```

```
Me.frmPersonSubform.Form.Recordset.AddNew

Me.frmPersonSubform.Form.FirstName.Value = strFirstName
Me.frmPersonSubform.Form.LastName.Value = strLastName
End Sub
```

Let's try a simple visual trace of the process to start to see if we can at least try to isolate where the problem occurred. Since clicking the button starts the process, and its name is cmdAddInfo, Access begins by telling VB to run the procedure named cmdAddInfo_Click. (In this version of VB, this represents the object and an event of the object, so this block of code is run. In VB.Net, the next generation of VB, this was improved and you can have any name attached to an event. We'll talk about VB.Net in a few more weeks.)

The code in our event procedure contains one line of code and one line that is commented. The commented line is not going to be processed, so that's not where the problem occurs. That leaves this line:

```
AddPerson GetPersonName( )
```

That consists of two parts: a procedure call (AddPerson) and a function call (GetPersonName). We know that AddPerson is a procedure call because (1) it doesn't show up in blue so it's not a VB keyword; (2) it is not a local variable because there is no Dim statement in the procedure; (3) it is not a "Class-level" (aka "Module-level") procedure because there is no definition for that at the top of the code block (normally appearing as "Private AddPerson As datatype," very similar to a Dim statement); and (4) there is a VB Sub with that name somewhere in the code window. Since it's a procedure call, it is allowed to have parameters, which we also see in the definition where we see Sub.

Note: You can quickly locate the definition of a name by right-clicking it and selecting "definition" (or selecting it and choosing menu View/Definition or Shift-F2.) Your cursor will move to the definition, which might be a Dim or a Sub or something else, and may be in this code window, another code window, or even external to your project.

We know that GetPersonName is a function because we can ask VB for the definition (as listed above), and it takes us to a VB Function line.

When a Sub call like AddPerson is processed, it must first collect the values to be passed. Since the value is a function, it must first process the function, then come back and call the Sub.

Looking at our function, there is really nothing special about it that would cause the name to get mixed up. The Dim just defines a variable; the next line splits the text on the Enter key and keeps only the first line; the last line returns the result.

So the problem must be somewhere in our Sub. The first few lines are Dim statements; they can't cause the problem since they merely set up space for the variables, but don't work with any data. Let's continue by exploring each line to quickly see if it's a candidate for our problem line.

```
strPersonNames = Split(PersonName, " ")
```

This just splits the name at the spaces and puts the pieces into an array. With "Abe Lincoln" we end up with Abe being the first item of the array (known in VB as strPersonNames(0)) and Lincoln being the second item (strPersonNames(1)). All that did was split the pieces. It didn't put anything in the wrong place or throw anything away.

```
iNameCount = UBound(strPersonNames)
```

This line uses VB's UBound function to get the upper boundary of the dynamic (non-fixed-size) array. In this case, it returns 1 since the highest item number is 1, as shown above.

```
Select Case iNameCount
```

This simply begins a block of implicit IF statements based on the value after the word Case. Let's skip the whole block for the moment and scan the rest to see if the problem occurs later. So scan down past the End Case since this ends the Case block.

```
Me.frmPersonSubform.Form.Recordset.AddNew
```

This line goes to me, the current class, which is the form in this case, then locates frmPersonSubform, the Person subform, then locates the Form on that subform, then finds the form's recordset (the set of records used to display the data), then adds a new blank record. This is simply creating a blank record for us to add information.

```
Me.frmPersonSubform.Form.FirstName.Value = strFirstName Me.frmPersonSubform.Form.
LastName.Value = strLastName
```

These go to the same subform and put something into the Value property of the FirstName and LastName objects (which are Text Boxes). At this point, we have to assume that the values should have already been calculated, since the problem has already occurred.

It looks like there's something wrong in the Case block, so let's go back to it. First, understand how the case works. The first line defines a value you want to match. In our case that was iNameCount, which was 1 in our problem case. Inside the Case block are a series of "Case" lines followed by some VB code that applies only to that case. For us, we see this:

```
Case 1
' Use as last name only
strFirstName = ""
strLastName = strPersonNames(0)
```

Looking at this, it's now pretty clear what happened. Since "Abe Lincoln" returned a value of 1, it left the first name blank and put the first item (which was the first name) into the last name field.

Digging deeper into the intent, it appears that 0 was supposed to mean no name given, so 1 was supposed to mean one name and 2 was supposed to mean two names. Further, the variable name iNameCount seems to confirm that.

So what happened? Well, it appears that iNameCount is wrong for our purposes. Instead of giving us 2 (as in two names) it gave us 1. Looking back at the place where it got its value, we see that it grabbed the upper boundary of our array, which was 1. Although we had 2 items in the array, since it's numbered starting at 0, 1 really means that we have 2 items or, basically, the number for the UBound is 1 less than what we want so we have to add 1 to get the right number. This turns out to be a very common problem where one reference is 0-based (starting with 0, like when you are measuring distance where you always start at zero), and the other reference is 1-based (which is typical when you're dealing with counting).

The natural solution here is to simply add 1 during our calculation. That's good, but there is one more less obvious problem. If you do that, then the lowest number you will ever have is 1, since the UBound determines the lowest array index, which starts at 0. Add 1 to this lowest number and now your lowest number is 1. So what? Well, we have a case for 0 that will never get used. Is that important? Maybe not, but where did that case come from?

The case for 0 arose from the idea that if there were no names, the routine should not save any names. However the Split function does not differentiate between one name (no delimiter found) and no names (still no delimiter). As such, we need to handle this special case if we want the 0 to be useful. So let's replace the iNameCount count assignment from this: "iNameCount = UBound(strPersonNames)" to this:

```
If PersonName = "" Then
    iNameCount = 0
Else
    iNameCount = UBound(strPersonNames) + 1
End If
```

This checks to see if the incoming parameter name, PersonName has nothing in it; if so, then use 0 to represent the number of names passed in. Otherwise, use the original calculation but add 1 to it.

One other note for those who like optimal code. The Split function on the line before this becomes useless if there is no PersonName passed and would therefore create a value for a variable that is never used. In this case, the more efficient thing to do is to move that line inside the above Else line. However, leaving it where it is means that it just slows your process down an unnoticeable tiny amount. Since this process is not repeated many times in a short period, we'll never notice the difference. However, if this were automated and running the process for thousands of names, that tiny amount of time might just add up and slow down your process enough to justify optimizing your code. We'll have to leave that discussion for a different article.

Run the form and Enter Abe Lincoln now and it should work. You may also want to go remove the previous, incorrect record with the last name "Abe."

Just to test a bit further, try adding "John Quincy Adams," and the application has another problem. It throws an error and lets you click on the Debug button to take you into the code on the following line in the Case Else section:

```
If InStr(strPersonNames(iNameCount), ".") > 0 Then
```

If you take your mouse and hover around various parts, you'll see that iNameCount shows 3 (because we have three names, which is correct) and strPersonNames(iNameCount) shows "<Subscript out of range>". Uh oh! Now what? Did we write more bad code? It turns out we just failed to update this when we updated our counter (iNameCount). Again, we have a numbering problem. iNameCount is 1-based but we're using it in a 0-based array. What we should have done when we added 1 in our previous calculation is that we should have subtracted 1 every time we use that within an array reference in a variable name.

You should have a yellow line, indicating that VB is waiting for you to respond to the problem. VB is nice in that it usually lets you change the code immediately when there's a problem like this. So change this to be iNameCount-1 inside parentheses and Continue (press F5 or menu Run / Continue) to see how it goes.

Well, now we see a problem here:

```
strLastName = strPersonNames(3)
```

Again, we get the same error. The problem here is that we were looking to see if the name contained a suffix title (like "Jr."). Since it didn't, we wanted to take the last word as the last name. Looks like we have the same problem of the wrong index, and if we look at the other part of the IF we have the same problem. What we should have had was this:

```

If InStr(strPersonNames(iNameCount - 1), ".") > 0 Then
    ' contains a dot so use previous as last name
    strLastName = strPersonNames(1)
Else
    strLastName = strPersonNames(2)
End If

```

By decreasing the array index (number inside parentheses) we should now have fixed this part of the problem and can again Continue.

This time no more errors. Switch back to Access and then back to the form and whoops! Still didn't really work! Again we got a first name in the last name field. One more time looking at the code, but this time let's only focus on the Case Else portion, since we know that's where the problem occurred.

```

Case Else
    If InStr(strPersonNames(iNameCount - 1), ".") > 0 Then
        ' contains a dot so use previous as last name
        strLastName = strPersonNames(1)
        ' check for . in first name
    Else
        strLastName = strPersonNames(2)
        ' try to find first name
    End If
    If InStr(strPersonNames(0), ".") > 0 Then
        ' dot found in first name so skip it
        strLastName = strPersonNames(1)
    Else
        ' use first word as first name
        strLastName = strPersonNames(0)
    End If

```

OK, we fixed the first IF. What's wrong with the second IF? If you can't figure it out, look carefully at what we're taking and where we're putting it. The first IF put the result into the strLastName variable. The other IF was supposed to put the result into the strFirstName variable. Oops! Sloppy programming because I rushed getting my article out a few weeks ago. Looks like I just copied and pasted but didn't update the parts.

Things like this are inevitable, but that's why it's also good to try various test runs to make sure it handles all reasonable cases before you release your code to your customers or users.

Fix the variable names and re-run this case. This time all is well.

One last thing: We should have added error trapping (On Error...) to better manage this problem. We'll add that next week.

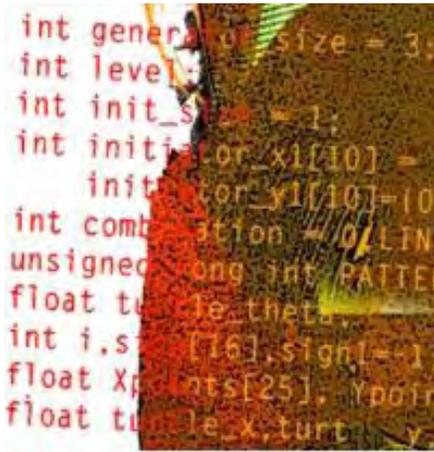
Also next week we'll try to solve the problem of putting a full name and address into the proper places. Stay tuned.

Rob has been in the computer industry for over 25 years and is currently a part-time teacher, offering classes in Excel, Access, Visual Basic, and a variety of other technical tools. He has loved *ComputerEdge* since 1990 and can be contacted at RSpahitz@Dogopoly.com.

Looking for a great boardgame? Grab a copy from DOGOPOLY.com (dogopoly.com) and have a dog-gone great time.



[Return to Table of Contents](#)



Programmer's Corner: Runtime Revolution

“Part II: Creating a User Interface in Runtime Revolution” by Wally Wang

Like most RAD tools, creating a user interface in Runtime Revolution involves a two-step process. First, you have to place a user-interface object on a window. Second, you have to customize that object, changing its properties.

Programmer's Corner

Continued from last week's "Part I: Programming in Runtime Revolution":

In traditional programming, you have to write code to create your user interface. In RAD (Rapid-Application Development) programming, you draw your user interface on a window and your compiler automatically takes care of making your user interface work.

Like most RAD tools, creating a user interface in Runtime Revolution involves a two-step process. First, you have to place a user-interface object on a window. Second, you have to customize that object, changing its properties. An object's properties simply define the size, position and appearance of that object on the screen.

In Runtime Revolution, a toolbox contains a palette of common user-interface objects, such as buttons, check boxes and scroll bars that you can arrange on a window to create your user interface.

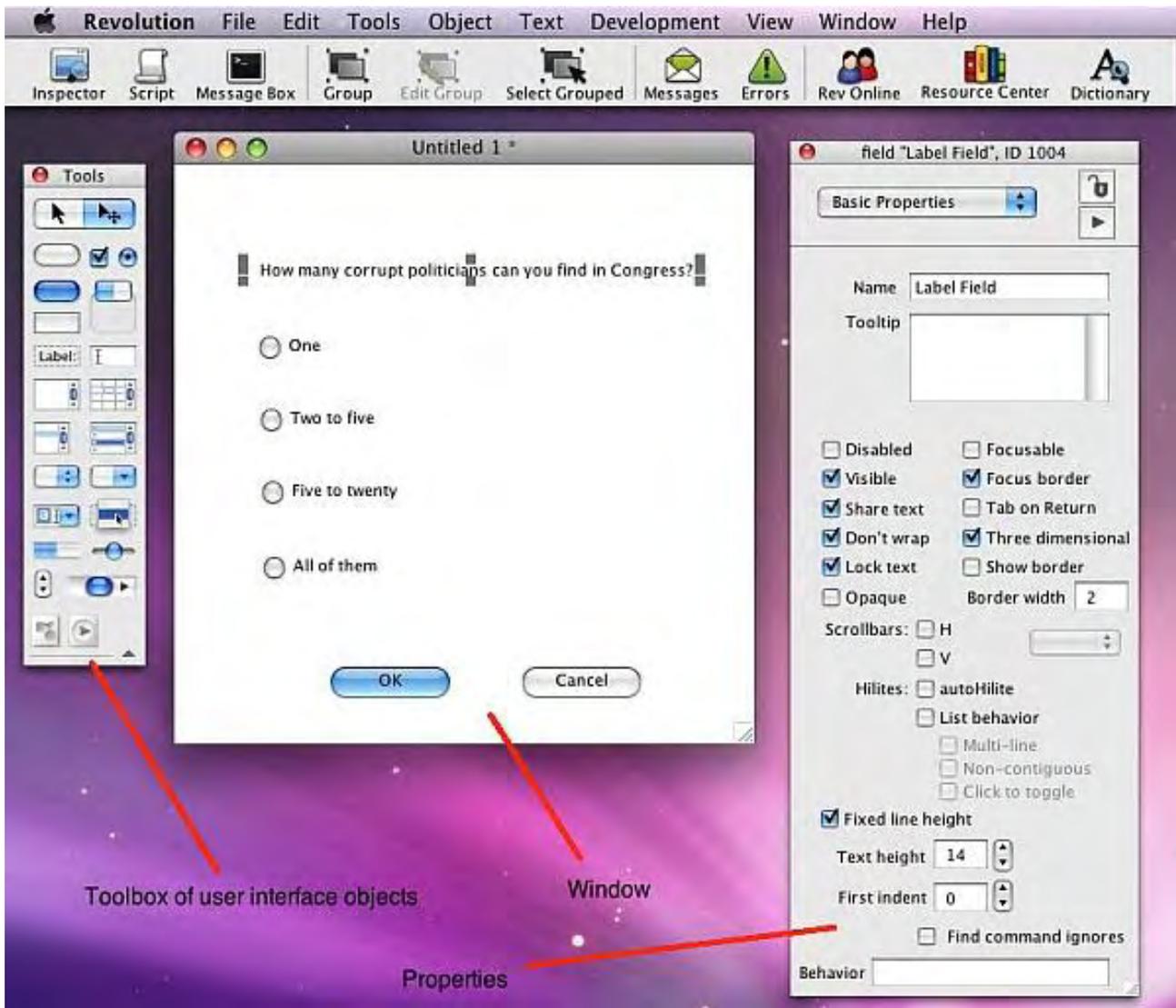


Figure 1. Runtime Revolution displays a toolbox of user-interface objects.

To place a user-interface object on a window, you can either drag a user-interface object off the toolbox and place it on a window, or you can double-click on a user-interface object on the toolbox and have Runtime Revolution display it on the window automatically.

Whichever method you use to place an object on a window, the next step involves customizing its properties. Common properties that most objects share include:

- Name: Lets you give each object a distinct name to identify that object.
- Label: Displays text on the object that's visible on the screen. For example, a button's label might be OK or Cancel.
- Width: The object's width.
- Height: The object's height.
- Location: The object's distance from the left and top of the screen.

Every user-interface object can display a Properties window, which lets you modify the appearance or behavior of that object.

Working with Windows

One major difference between Runtime Revolution and other RAD tools like Visual Basic is how Runtime Revolution handles windows. A window is simply a rectangular area where you can place one or more user-interface objects.

Simple programs, such as a calculator, might consist of a single window, while more complicated programs, such as a word processor, might consist of dozens of different windows.

In Runtime Revolution, a window is called a "card" and multiple cards of identical sizes are called a "stack," which harkens back to Runtime Revolution's HyperCard roots. (You can even take old HyperCard stacks and run them in Runtime Revolution.)

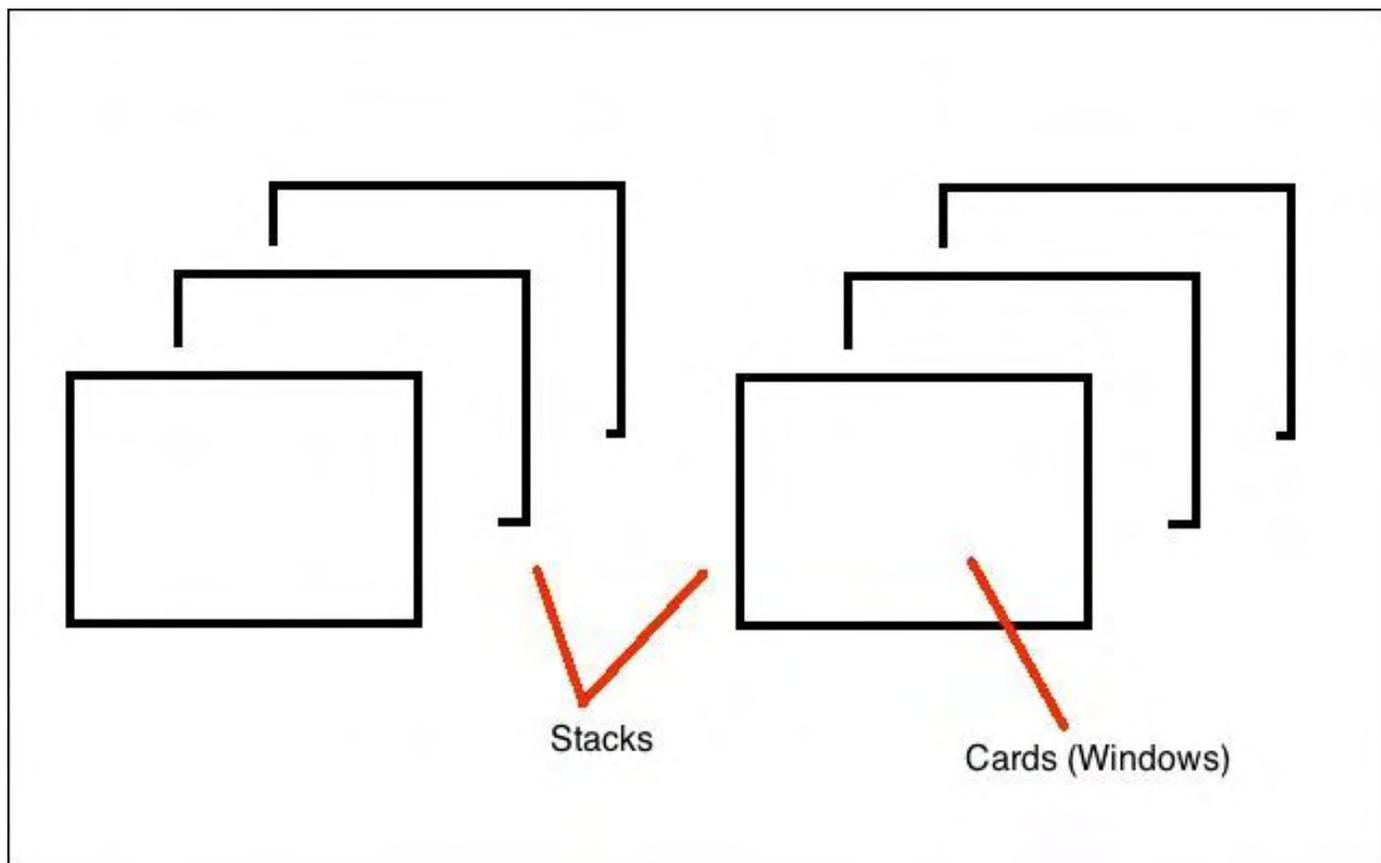


Figure 2. A Runtime Revolution program consists of multiple stacks and cards.

Every Runtime Revolution program consists of a main stack with one or more cards. Cards in a stack can share the same background objects, such as buttons or a background image.

If you want to create windows of different sizes, you have to create substacks. You can have as many substacks as you wish, but only one main stack. By forcing you to organize your windows into cards and stacks, Runtime Revolution helps you organize the information stored on your cards.

For example, one substack might consist of nothing but cards that display error messages. Another substack might contain cards that only accept data from the user. By organizing similar cards in stacks, you can keep the different parts of your program separate and (hopefully) organized.

Making User-Interface Objects Work

After creating all the windows (cards) you need for your program and placing user-interface objects on each window, the final step is to write code to make your user interface actually do something.

The three common types of actions you'll need to make a user interface work are:

- Make your program do something (such as when the user clicks an OK button).
- Retrieve data from the user interface (such as when the user types in a number).
- Display data on the user interface (such as when your program displays an error message on the screen).

RAD programming is often called event-driven programming because your program responds to events that occur on the user interface. Any time the user does something to the user interface, such as click the mouse over an object, that object sends a message to the program that says, "Hey, the user just clicked a button."

Although Runtime Revolution recognizes when the user does something to a user-interface object, you must write instructions, called a script, that tell your program exactly what to do when a particular event occurs.

You might write one script that tells your program what to do when the user clicks on a button, and a second script that tells your program what to do when the user presses a certain key or clicks on a different user-interface object.

If you want a user-interface object to do something when the user clicks on it, you'll need to write a script. This script defines two things:

- What event to respond to
- What to do

For a button, the most common event to respond to is called `MouseDown`, which tells your program, "Hey, do something if the user clicks the mouse." The simplest script consists of a single command, `Quit`, such as:

```
on mouseup
    quit
end mouseup
```

Every user-interface object can respond to its own `MouseDown` event. So if the user clicks on a button labeled `Quit`, it might run the above script, but if the user clicks on a button labeled `Make Noise`, it might run a completely different `MouseDown` script, such as:

```
on mouseup
    beep 3
end mouseup
```

This script simply tells the computer to beep three times. If you wanted your program to beep three times the moment the user moved the mouse over a user-interface object, such as a label that displays text, you could use this script:

```
on mouseenter
    beep 3
end mouseenter
```

By creating different scripts to handle different events, such as clicking the mouse or moving the mouse over an object, you can make your user interface respond to the user.

Retrieving and Displaying Data From a User-Interface Object

Oftentimes the user will need to enter some data into your user interface, such as typing a name or number in a text box. You don't need to write any code to let the user type anything into user-interface objects, but you do need to write code

to retrieve whatever data the user typed.

To retrieve data, you need to know the name of the user-interface object that contains the data and then get that data using the Get command, such as:

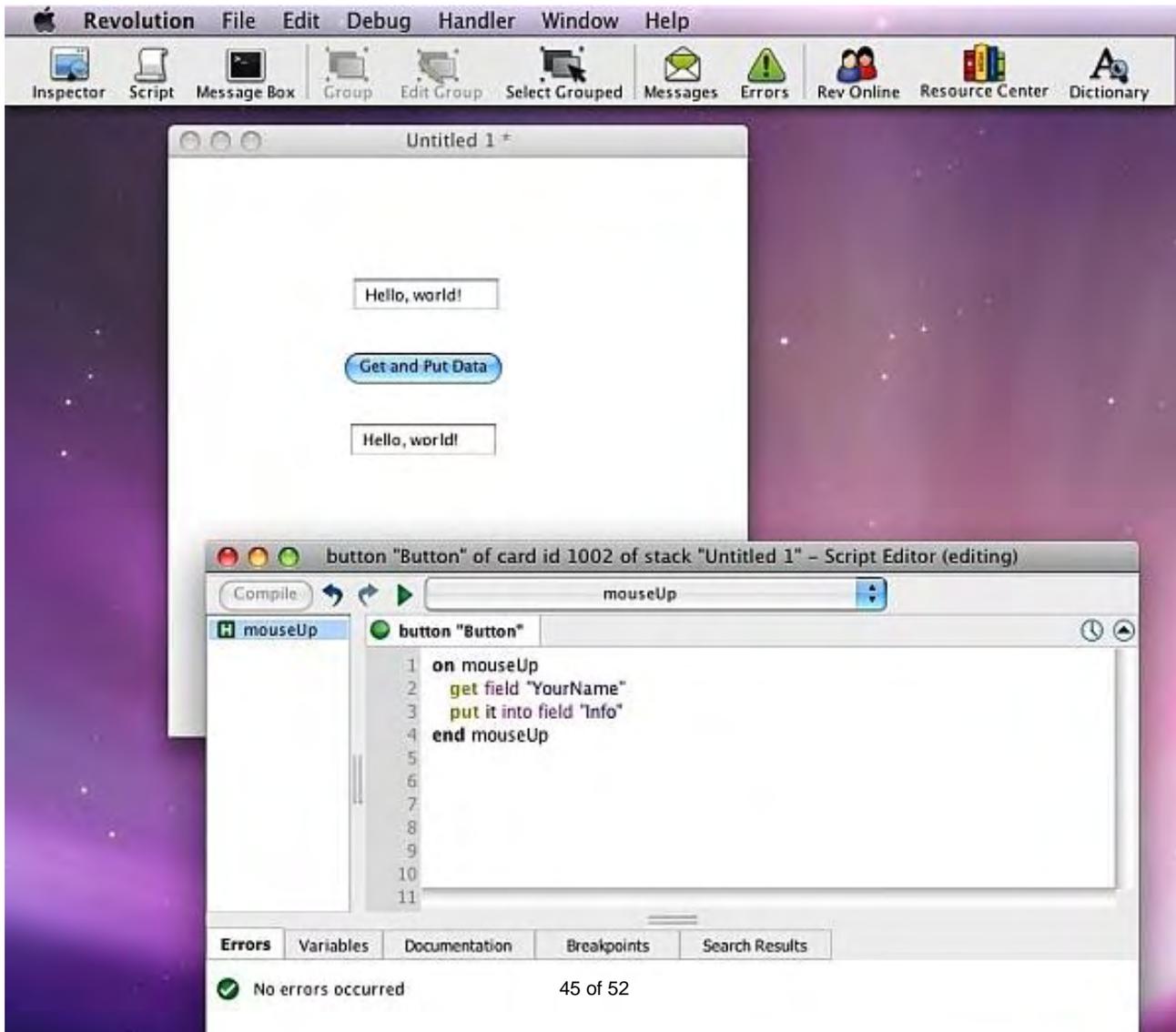
```
get field "YourName"
```

This command tells your program, "Get the data stored in a field named 'YourName'." When the Get command retrieves data, it stores that data in a temporary variable called "it." If you wanted to display this data in another field, you could use this script:

```
on mouseUp
  get field "YourName"
  put it into field "Info"
end mouseUp
```

This script runs when the user clicks on the user-interface object, such as a button. The first line tells the computer to retrieve the data stored in a field called "YourName." This retrieved data gets stored in the temporary variable named "it," although this code doesn't explicitly state this.

The second line tells the computer to take the variable stored in the "it" variable and put it into the field named "Info." Basically, this short script lets the user type data in one text field and display it in another after clicking a button.



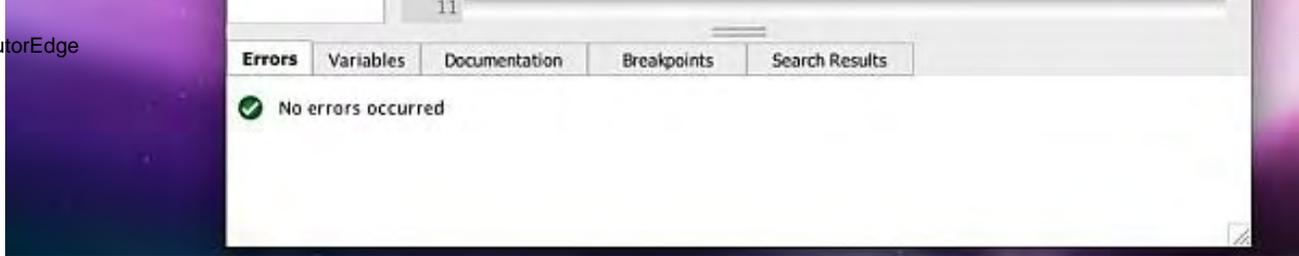


Figure 3. How the script retrieves and displays data.

The main idea to making your user interface work is to customize the objects that make up your user interface, and then write code to make that user interface do something, retrieve data, or display data. In the last article of this series, I'll show you more about how the Transcript language works to do common actions, such as mathematical calculations or make decisions.

Next week: Part III: Coding in Transcript

In the early days, before Wally became an Internationally renowned comedian, computer book writer, and generally cool guy, Wally Wang used to hang around The Byte Buyer dangling participles with Jack Dunning and go to the gym to pump iron with Dan Gookin.

Wally is responsible for the following books:

- Microsoft Office 2007 for Dummies (www.amazon.com/gp/product/0470009233?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0470009233)
- Beginning Programming for Dummies (www.amazon.com/gp/product/0470088702?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0470088702)
- Breaking Into Acting for Dummies with Larry Garrison (www.amazon.com/gp/product/0764554468?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0764554468)
- Beginning Programming All-in-One Reference for Dummies (www.amazon.com/gp/product/0470108541?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0470108541)
- Steal This Computer Book 4.0 (www.amazon.com/gp/product/1593271050?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1593271050)
- Visual Basic Express 2005: Now Playing (www.amazon.com/gp/product/1593270593?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1593270593)
- My New Mac (www.amazon.com/gp/product/1593271646?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1593271646)
- My New iPhone (www.amazon.com/gp/product/1593271956?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1593271956)
- Strategic Entrepreneurism with Jon Fisher and Gerald Fisher (www.amazon.com/gp/product/1590791894?ie=UTF8&tag=the15minmovme-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1590791894)

Every Saturday morning from 9:00 am - 10:00 am in San Diego, you can hear Wally with fellow co-hosts Dane Henderson and Candace Lee, on the radio show CyberSports Today (cybersportstoday.com/), which covers the video gaming industry on ESPN Radio 800 AM. Wally covers the military history side of the video game industry.

When not performing stand-up comedy or writing computer books, he likes to paper trade stocks with the video game Stock Reflex (www.plimus.com/jsp/download_trial.jsp?contractId=1722712&referrer=wwang), using the techniques he learned from a professional Wall Street day trader.

Wally can be reached at wally@computoredge.com and at his personal web site (www.wallacewang.com/).

[Return to Table of Contents](#)



ComputerQuick Reviews

News and Reviews from Readers and Staff

ComputerQuick Reviews

“Send us your thoughts about your favorite hardware, software and Web sites.” by

ComputerEdge Staff

We want readers' opinions about hardware, software, and Web sites. Share your thoughts and ideas!

We Want Your Opinions About Hardware, Software and Web Sites

Over the years, *ComputerEdge* has had great input from our readers. In particular, people have submitted short reviews of equipment, software and Web sites that they really like. In some cases readers have offered tips (such as avoiding flakes on Craigslist). ComputerQuick Reviews is our column dedicated to highlighting those things that you most like and want to recommend to others. The problem is that if this column doesn't appear, it becomes forgotten and less likely to receive input from you.

We have decided to include this feature in every issue as a reminder that this is your magazine—even if we don't have any new reader reviews. If you would like to see the type of reviews that we have run in the past, then check out ComputerQuick Reviews (webserver.computoredge.com/sitemap.mvc?feature=Columns&columnedcode=persrev&column=ComputerQuick%20Reviews) in the *ComputerEdge* Site Map. You will find that they are quite varied. We would like to see more. Consider this column a gentle prod saying that we would like to hear from you.

You can send us an e-mail at ComputerQuick Reviews Submissions (ceeditor@computoredge.com).

The purpose of this column is to give our readers an opportunity to express their opinions about products and services that they have found particularly useful. If you have had experience with hardware, software or a Web site that made you say, "This is really great! I want to tell everyone about it," then this is a good place to do it. While we do want to post warnings, we are not interested in slamming the obscure bad products, because there are too many of them. We would like to hear about those things that you would recommend to your best friend. The only caution is to please use proper capitalization (do not use all caps) and complete sentences. If it takes us too long to edit the piece, it may be a long time before it's published on this site.

Please send your personal reviews to ComputerQuick Reviews Submissions (ceeditor@computoredge.com).

ComputerEdge always wants to hear from you, our readers. If you have specific comments about one of our articles, please click the "Tell us what you think about this article!" link at the top or bottom of the article/column. Your comments will be attached to the column and may appear at a later time in the "Editor's Letters" section.

If you want to submit a short "ComputerQuick Review", or yell at us, please e-mail us at ceeditor@computoredge.com.

[Return to Table of Contents](#)

EdgeWord: Technology to Change the Way You See Things

edge **WORD**

“Handheld devices are evolving to save battery life.” by Jack Dunning

The screen-power problem is being addressed by new technologies, such as IMOD, MEMS and OLED.

The primary problem with most handheld devices—whether cell phones, GPS or other portable device—is the battery dying. This is also true for laptop computers. While battery technology seems to slowly improve, the power demands on the devices continue to grow. For example, in a scenario I outlined not too long ago, an iPhone was used with a golfing application and its GPS capability to determine the direction and distance to the next hole. Definitely a useful application for any golfer. The problem was that the iPhone application needed to be turned off when not in use. Leaving the application on would quickly drain the battery. It's not easy to recharge a battery in the middle of the course.

One of the primary culprits for draining battery power is the viewing screen. Most of today's handheld devices and laptop computers use liquid crystal displays (LCD) for a monitor. Because LCDs do not emit light, a backlight is required to see the screen—especially at night and in bright light. This backlighting draws a good bit of power. This is the reason that most cell phones will quickly turn off the light when not in use. This is not a problem when the device is being used as a phone, but once the phone became a computer, the need for the activated screen and the associated power drain became more necessary.

There are a couple of ways that the screen-power problem is being attacked. One is called Interferometric modulator display (IMOD), which uses micro electromechanical systems (MEMS) technology to build highly reflective displays. (Don't worry, I'll explain.) Qualcomm has given the marketing name Marisol to its endeavors in this area. The second primary thrust in new display technology is Organic Light Emitting Diodes (OLED). This technology is being pursued by most of the high-tech consumer electronics companies based upon licensing from Kodak, which holds some patents supplying materials and expertise.

In the IMOD approach used by Qualcomm, tiny sets of two mirrors (one is half-silvered to allow light to pass through it) are used to reflect the natural light in a way that creates the needed pixel. There are two positions for each set of mirrors, reflecting either a color or no reflections at all (black). See Figure 1. This creates a highly reflective display that can be viewed in any light without any backlighting. In fact, the brighter the daylight, the brighter the display appears. It is faster than LCD technology and draws much less power because no backlight is required, and power is applied only to change the state of a pixel.

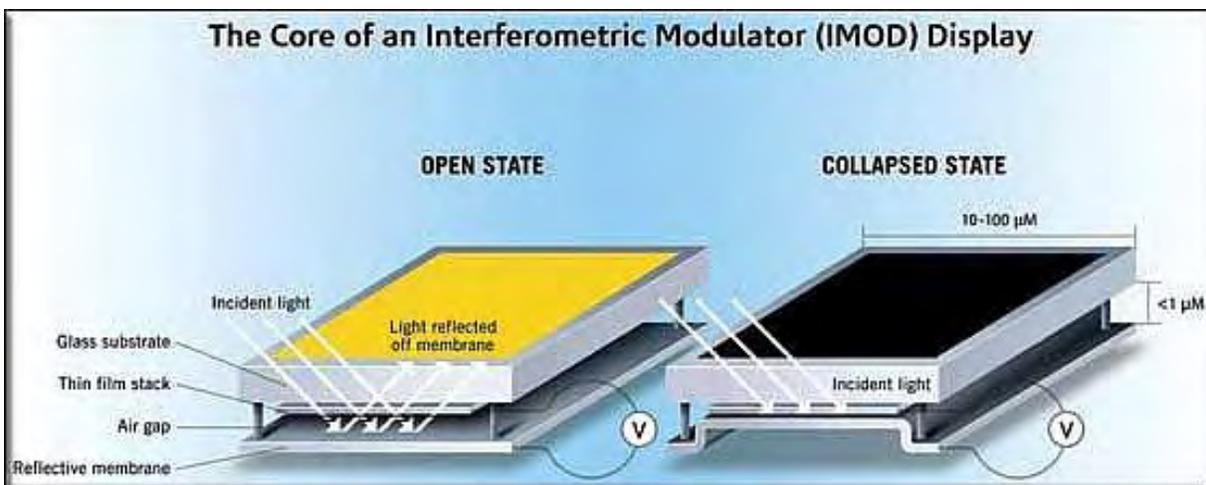


Figure 1. How IMOD works. Simple, huh?

There are a number of limitations. In particular, the problems of producing these tiny mechanical devices—one mirror

in each mirror pair actually moves—has restricted their use to small devices, with cell phones currently considered the probable upper limit. There are numerous problems with building multicolored displays, since a set of mirrors is needed for each, mixing wavelength and reflective intensity. Each pixel needs to be a matrix of mirrored pairs. If there is no ambient light, then a front light is needed, although the display seems to work well in even very low-light situations.

While IMOD displays have potential—especially in small applications that do not require full color—most of the industry seems to be betting on OLED technology. OLED technology is organic in that it uses polymers to produce the diodes. When a charge is applied to the OLED material, light is emitted (see Figure 2). Since the material is the source of light, no backlight is required. Nor is there a limited viewing angle, as is found with LCDs. A person can view an OLED display at almost a 90-degree angle and still view the screen.

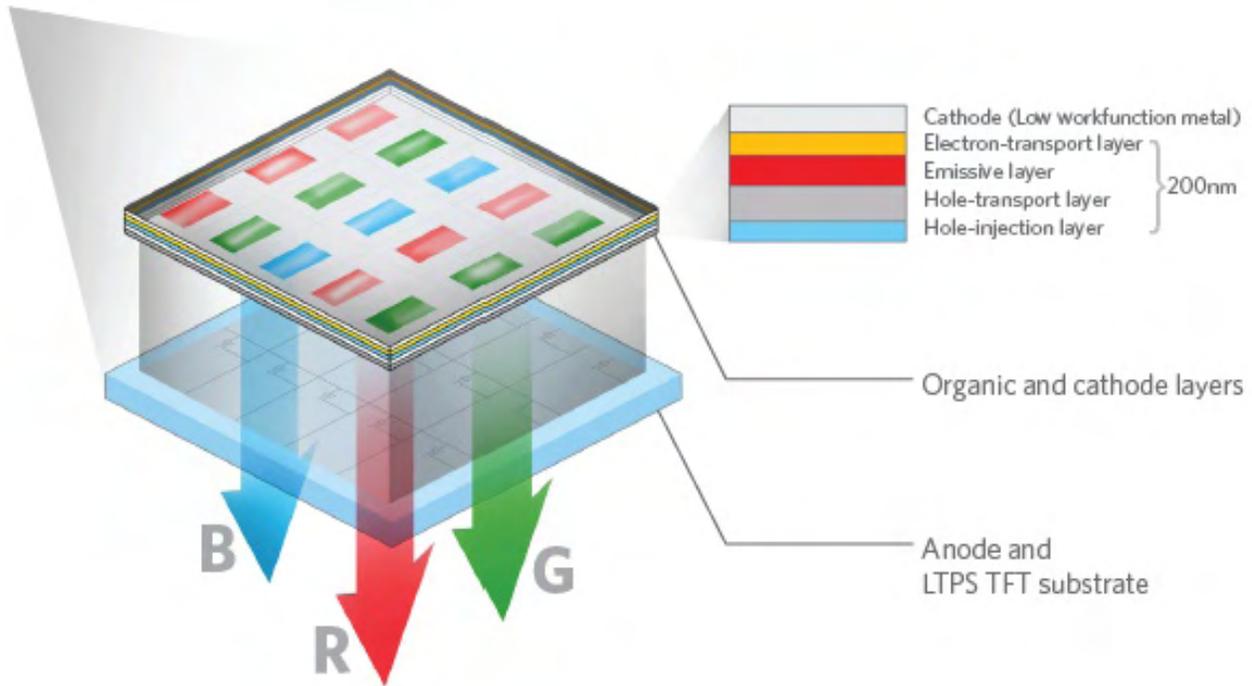


Figure 2. Different OLED material is used for each color diode, blue (B), red (R) and Green (G).

This technology is most likely the next wave of displays for televisions, computers, cell phones and almost any technology that needs a screen. It draws very little power and produces astonishingly bright images. It can be made into ultra-thin layers and may be put on flexible surfaces (T-shirts?). (For an example see, "Sony's Flexible OLED" on YouTube (www.youtube.com/watch?v=NcAm3KihFho&eurl=http%3A%2F%2Fwww%2Emnn%2Ecom%2Ftechnology%2Fgadgets%2Delectronics%2Fblogs%2Foled%2Dtechnology%2Dlights%2Dup%2De%2Dpassport&feature=player_embedded)). This is really cool!

Kodak introduced a digital OLED picture frame this year (see Figure 3). It has a seven-inch screen and costs about \$900. It's expected that the price of OLED displays will drop considerably over the next few years, since it is not an expensive manufacturing process—similar to printing.



Figure 3. Kodak OLED Wireless Frame.

The main drawback for OLED devices has been the relatively short life of screens due to degradation of the organic compounds—in particular, the blue emitting material. Companies are developing ways to deal with this problem. DuPont seems to have made some major strides in producing OLED materials that will last 100 years (www.reliableplant.com/article.aspx?articleid=18503&pagetitle=DuPont+OLED+technology+achieves+1+million+hours+lifetime)—at least for the green. Currently, the red and the blue materials are limiting life to about 100,000 hours (over four years of continuous operation).

OLED technology can be used in big-screen televisions as well as cell phone screens. Companies are starting to introduce a variety of devices (televisions, computer monitors, etc.), although most are currently expensive. There are also plans in the works to use OLED technology for low-power lighting. It seems OLED may be the technology that will replace LCDs. The development of OLED in the coming years is certainly something to watch.

Jack is the publisher of *ComputerEdge* Magazine. He's been with the magazine since first issue on May 16, 1983. Back then, it was called *The Byte Buyer*. His Web site is www.computoredge.com. He can be reached at ceeditor@computoredge.com

[Return to Table of Contents](#)



Editor's Letters: Tips and Thoughts from Readers

“Computer and Internet tips, plus comments on the articles and columns.” by ComputerEdge Staff

"Access and FoxPro?," "A-Squared," "Using Windows 7," "Checking E-Mail for Scams," "Great Windows 7 Article!"

Access and FoxPro?

[This letter is in regard to Rob Spahitz's June 19 column, "Rob, The ComputerTutor Does Microsoft Access."]

Thank you, I've enjoyed your articles for a long time.

It occurred to me today seeing the Access article to ask: Is Access a newer version of the old FoxPro database program?

Thanks,

-Paul Bingham, Littleton

A-Squared

[This letter is in regard to Jack Dunning's May 29 article, "The Best Free Antivirus Programs."]

I had a friend recommend A-Squared (www.emsisoft.com/en/software/free/) mainly because it could be run as a portable app. I tried it, and it is awesome. I was a Norton user and fan for years, but it just got to be so expensive, that when I upgraded to a Mac I couldn't justify the cost to protect the Win partition.

-No Name Given

Using Windows 7

[The next letters are in regard to Jack Dunning's June 26 article, "Using Windows 7."]

Well researched and compiled—terrific input. Thanks!

-Lynn Manning Ross, San Diego

I have enjoyed these articles on Windows 7. They have been very informative and clear/easy to understand. This article is keeping me up to date with the changes in Windows 7. Thanks for taking the time to write this article, as it saves me a lot of time keeping up.

-Scott Brand, Salem, CT

Checking E-Mail for Scams

[The following letters are in regard to Jack Dunning's June 26 Edgeward column, which offered malware tips.]

An excellent method of checking on any incoming e-mail is to click on File, then on Properties, and you can then examine the detailed description of the source. If anything appears out of order, delete the message. Personally, I do this check on every incoming message whose source is not immediately apparent and acceptable—a habit that is more successful than looking for a potential problem. Check everything except the known valid e-mails.

-Allen Gates, San Diego, CA

Another technique I've found useful is to carefully read the whole thing looking for bad English. Almost every phishing e-mail I've ever come across has had some—even when the scammer appears to be an English speaker!

In the screenshot given with this article, there is the phrase "latest version of the Microsoft Outlook / Outlook Express." While I'm well aware of the dismal state of English in America, Microsoft's English tends to be pretty perfect. That makes the "the" before "MS Outlook / Outlook Express" a dead giveaway, in my estimation. And I think the spaces on both sides of the slash are also a non-U.S. convention. Of course, maybe you have to be a pretty good grammar cop to catch those, but if you happen to be or know one, that's an extra technique you can use to ferret out weasels.

(P.S. In this case, the scammer appears to be a French speaker (unexpected article with noun; spaces around slash.)

-Werner Maurer, Rosarito Beach, Mexico

Great Windows 7 Article

[This letter is in regard to Jack Dunning's June 26 article, "Windows 7: New and Improved Programs."]

This is a great article, and the whole series of articles concerning Windows 2007 has been great. They have kept me in the loop of new technology for myself and work. I have been able to update others. Thanks for the time that went into this and the other articles. It is appreciated.

-Scott Brand, Salem, CT

ComputerEdge always wants to hear from you, our readers. If you have specific comments about one of our articles, please click the "Tell us what you think about this article!" link at the top or bottom of the article/column. Your comments will be attached to the column and may appear at a later time in the "Editor's Letters" section.

If you want to submit a short "ComputerQuick Review", or yell at us, please e-mail us at ceeditor@computoredge.com.

Send mail to ceeditor@computoredge.com with questions about editorial content.

Send mail to cwebmaster@computoredge.com with questions or comments about this Web site.

Copyright © 1997-2009 The Byte Buyer, Inc.

ComputerEdge Magazine, P.O. Box 83086, San Diego, CA 92138. (858) 573-0315