

Isys Information Architects

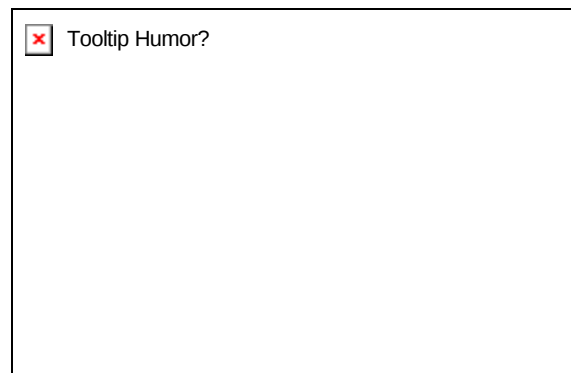
Making information usable

Interface Hall of Shame

- Interface Stupidity -

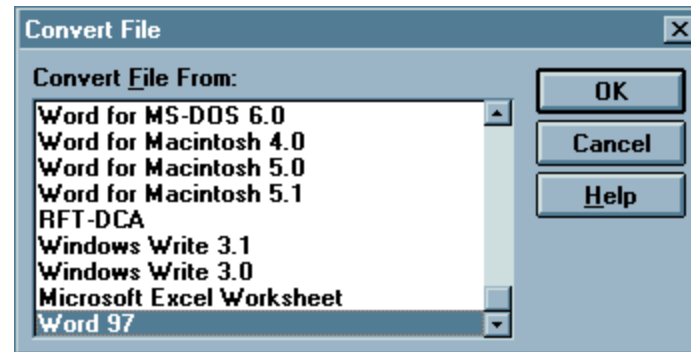
Despite remarkable advances in technology, many computer applications still appear to have the I.Q. of a toaster. Many computer applications interrupt the user with silly questions, require the user to explicitly specify obvious selections, fail to remember even the simplest of things, and make inane, unnecessary statements. Here are a few examples:

Last updated 19-September-1999



Chris Gumprich sent us this image from Microsoft's *Windows Media Player*. It would seem that the developer wanted to be certain that he or she provided a tooltip for every control, regardless of whether or not the message was needed or helpful.

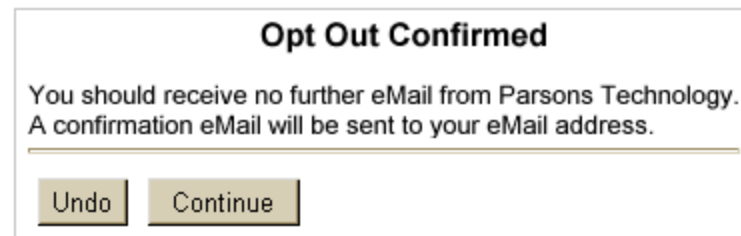
OK, make mine a Chivas on the rocks...



Microsoft Word 6.0 when asked to open a document from an unknown version of *Word* displays the above message. *Word*'s conversion utility seems to be asking, "I think it's a Word '97 document, but it might be one of these other types. What do you think?".

How the #\$\$%@& would I know!

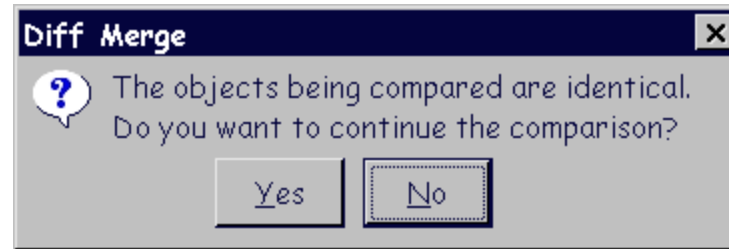
This is the result of a confirmation-happy programmer. The conversion utility knows exactly what type of file it is (the raw file contains two explicit references to the type of file), yet the program wants the *user* to confirm the program's ability to read these references. The user, on the other hand, unless he or she created the file, has absolutely no knowledge of the file type. By needlessly asking the user, the program needlessly creates uncertainty and an opportunity for the user to cause an error.



Mark McIntyre sent us this image of a message he received after unsubscribing from an unwanted mailing list. As Mark noted:

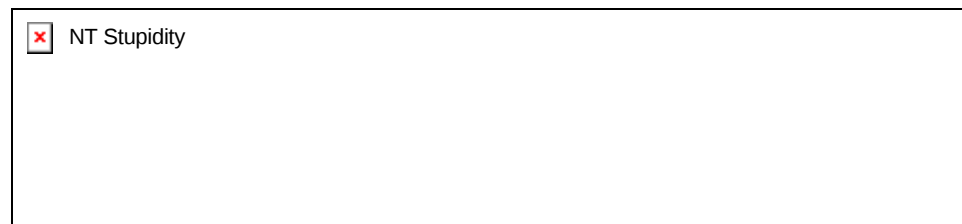
I couldn't help but laugh out loud at the complete idiocy of the message.

Neither could we.



Kate Adams sent us this example of interface stupidity from *ClearCase*, a source-code control system from Rational Software. Given the result, the question strikes us as rather *irrational*.

Jacek Wojdel sent us this image of a message he received from an NT workstation:

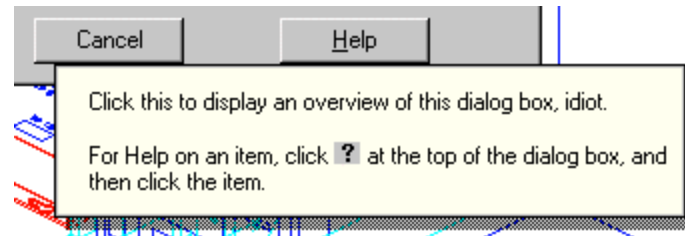


Duh!

Cancel just about any function in Quicken's *Turbo Tax*, and the program will thoughtfully tell you that you cancelled the function. Duh! Such stupidity is somewhat troubling for an application that we're supposed to trust to represent us to the Internal Revenue Service.



This revealing example of context-sensitive, though hardly user-sensitive, help in *AutoCAD Mechanical* was provided to us by visitor **Jerry Albro**:

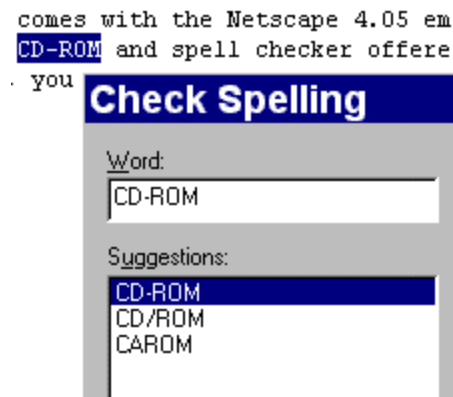


While some may regard the message as humorous, we find it particularly contemptuous.

We are reminded of a recent drugstore advertising campaign with the motto, "treat every customer as you would want your mother to be treated." That *AutoCAD* allowed this message to slip into its final product is truly shameful.

Update: We've received word from an individual at AutoDesk that the author of the message considered it a joke that would never be seen by the users. As a result of the joke, the employee is now a *former* employee, and AutoDesk has sent thousands of letters of apology to its customers. Who's laughing now?

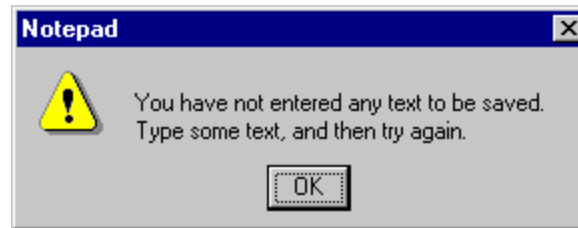
This example of the wonders of artificial intelligence in *Netscape Communicator* was provided to us by visitor **Simon Wilkinson**:



Just thought I would write in to inform you about the very unintelligent spell checker which comes with Netscape Communicator. When checking the word 'CD-ROM' the spell checker offered as a suggestion to this 'wrong' word, you guessed it 'CD-ROM'.

(For the record, it's a letter 'O', not a zero).

This image from Microsoft's *Notepad* was provided to us by **Todd Barlow**:

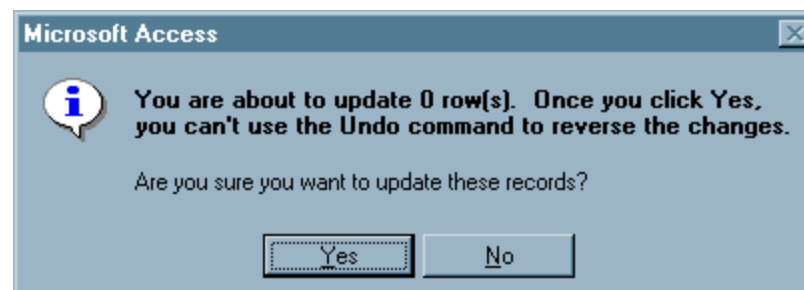


Here is my nominee for the Hall of Shame. This message appears when I try to save an empty Notepad file. You may ask "Why save an empty file?" This is the way I work. I open Notepad. I think about what I'm going to write. I give the file a name. I start writing. I was able to do this in Win 3.1.

Evidently, my work habits need improvement and Microsoft has taken responsibility for telling me.

François Gouget was one of many others who also nominated this particular *Notepad* feature. François also pointed out that the operating system itself has no problems creating empty text files:

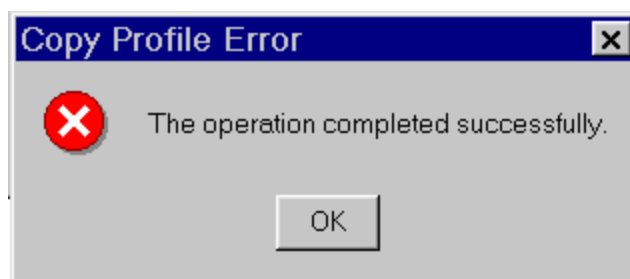
...did you notice that notepad will not let you save an empty file? But using the "New.../Text Document" contextual menu in Explorer will create an empty (0 bytes) text file that notepad will then open without any problem. So why have this arbitrary restriction?



Well actually, no, I do not want to update the records that wouldn't be updated anyway.

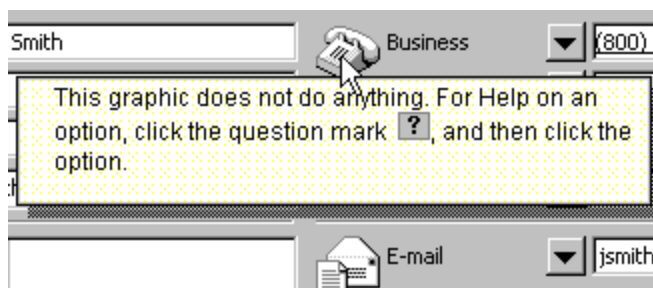
Duh! This informative message is provided by Microsoft's *Access 95* in response to the user's having executed a query that has ... no effect on the data.

This interesting feature in Microsoft's *NT Operating System* was provided to us by visitor **Jerry Albro**:



The message was generated after Jerry copied his NT user profile in System Properties dialog. Since when is success an error?

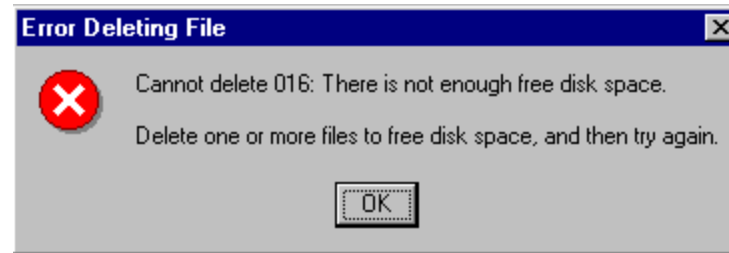
This feature in Microsoft's *Outlook 98* was provided to us by visitor **James Jarrett**:



I ran across this when I right-clicked on a bitmap in the Contact Editor in Outlook 98. Notice that it tells you that the bitmap doesn't do anything and then refers you to a question mark in a gray box that doesn't exist! I think they are trying to refer you to the question mark in the yellow talk balloon in the toolbar, but I'm not sure.

(We're still chuckling...)

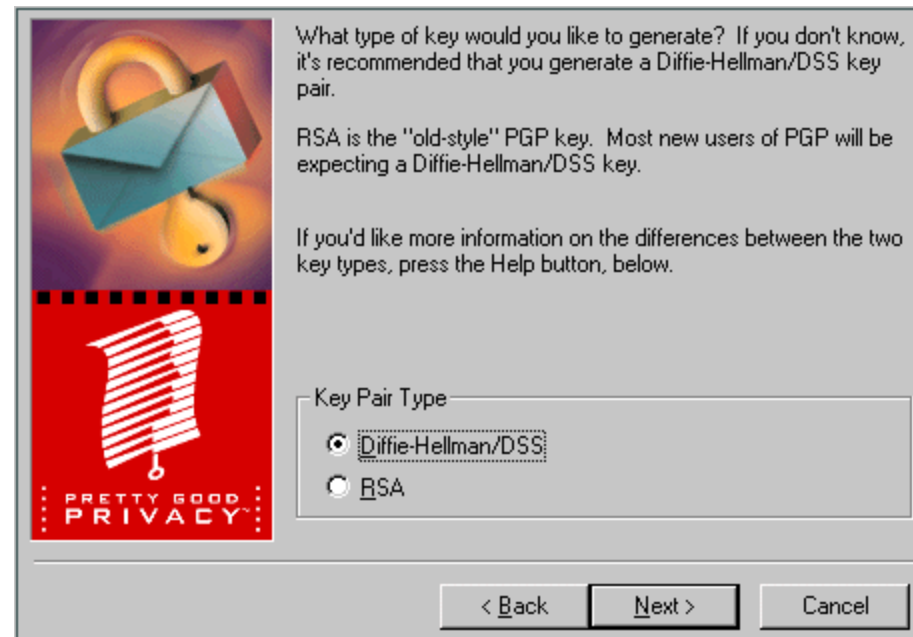
This gem in *Windows95* was provided to us by visitor **Roy Child**:



I came across this message when trying to delete files from a nearly-full hard drive in Windows 95. This has got to be the #1 stupidest error message I have ever seen.

Perhaps this an indication of Microsoft's interpretation of the term "Artificial Intelligence".

This image from *Pretty Good Privacy* was provided to us by visitor **Emanuela Moreale**



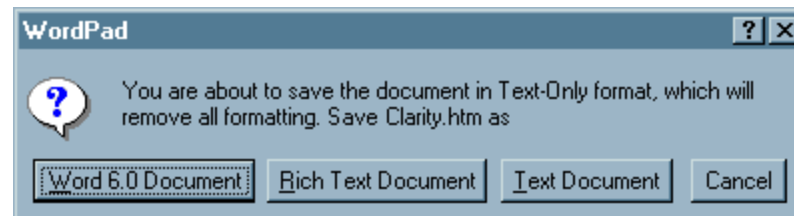
While PGP's designers thoughtfully provided a Wizard to assist the user in generating an encryption key, and thoughtfully provided a very professional and rather pleasing graphic image, they forgot to provide the Help button referred to in the text of the dialog. Cryptic indeed.

This example of useless messages was provided to us by visitor **Lin Ziyuan**.

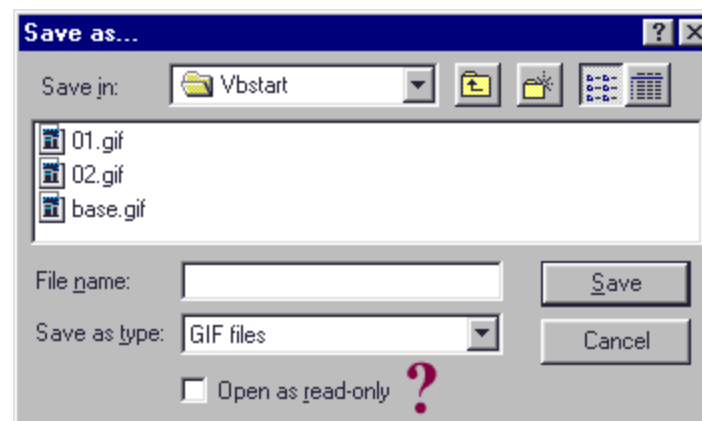


I'm writing in to complain about the new feature in the newest version of Microsoft's browser, Internet Explorer. Everytime when you download a file, a 'Download Complete' message box pops up. This is particularly irritating if you've got a lot of files to download. Why does IE4 need to report that it has completed downloading anyway?

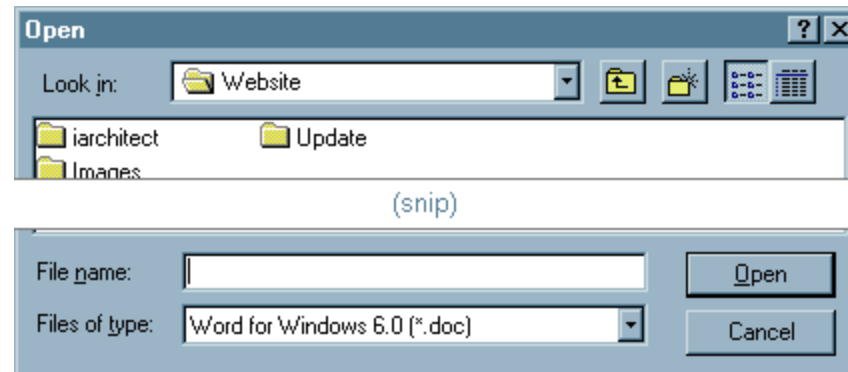
It doesn't, it's just being stupid. Despite all of their purported usability testing, Microsoft did not realize that this was a problem when they released *IE 4.0*. The problem could be resolved by downloading a 15MB fix to *IE 4.0*, which, among other fixes, provided an option to turn off this unnecessary notification. In later versions of *IE*, the option is provided, although the default is to display the message.



Microsoft's *WordPad* can be infuriatingly stupid. This message was presented after the user has opened an existing text-only document, made some changes (none of which involved formatting changes), and attempted to save it. The user's first response to the message is one of alarm ('REMOVE ALL FORMATTING?!'), but after seeing it the first several hundred times, the response becomes one of anger: 'If I wanted to save it as a different file type, I would have selected **Save As**, and specified a different file type!' This occurs each time you save the document even if you haven't closed it between changes.



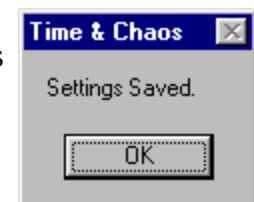
This option in this particular dialog in the *GIF Construction Set* could leave quite a few users scratching their respective heads. It might even cause some of them to immediately cancel the dialog, thinking that they selected File Open rather than File Save.



We have never opened a *Word* document in Microsoft's *WordPad*. Yet every time we attempt to open a document in *WordPad*, it assumes that we want to open a *Word* document. We use *WordPad* to edit our HTML documents (such as the one you are reading now), but *WordPad* does not provide an "HTML documents (*.htm)" file-type option, nor does it provide the ability to add it to the list of file types (which really would have been useful).

While hoping for that feature might be expecting too much, what we would really like to see is that *WordPad* be given the ability to remember the last specified file-type. Thus, if we specified "Text Documents (*.txt)", the last time we used *WordPad*, the File Open dialog would default to the "Text Documents (*.txt)" file type the next time we used *WordPad*. We don't believe that asking a computer to remember something would be asking too much.

Time & Chaos is a really useful PIM application. However, it has the unnerving tendency to interrupt the user with useless statements. The following message is displayed, and requires a response, whenever the user has made changes to the program settings. It's almost as though the program is seeking a pat on the head.

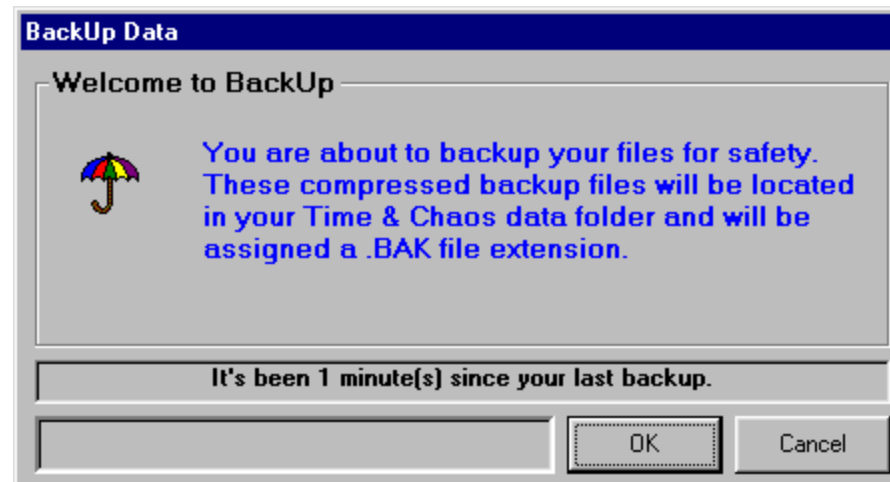
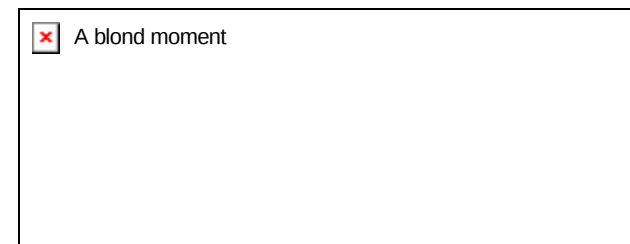


If there is any benefit to this intrusion, perhaps we should display similar messages for other command selections:

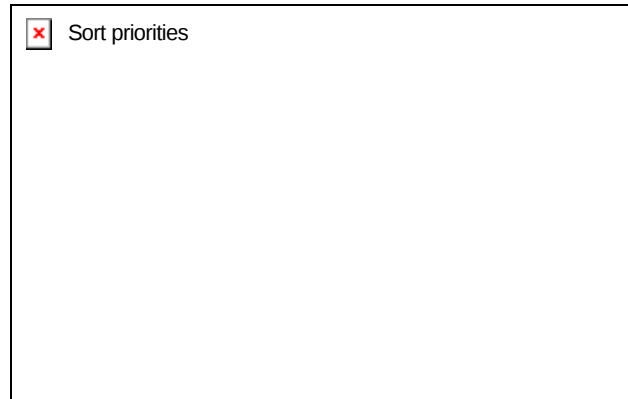
- "The selected text has been copied - OK?"
- "You have selected Print - OK?"
- "Your page has been printed - OK?"

We'll stop now - OK?

When you exit the event scheduling program *Automate Pro*, the program thoughtfully reminds you that doing so will prevent all scheduled tasks from being performed. The fact that it also warns you even when you have no tasks scheduled caused us to wonder just how much we should rely on the program.



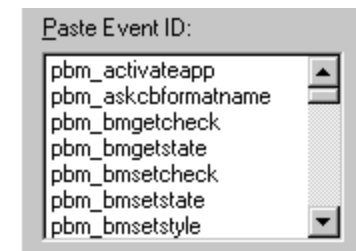
Another example from *Time & Chaos*. Whenever the user exits the program, this dialog is presented, even if no changes have been made. A better, less intrusive, and more intelligent method would be to automatically perform the backup when necessary.



PowerBuilder has a penchant for making software development a truly inefficient process. Throughout the application, when a list of variables or properties is presented to the user, the list is sorted by data type (e.g., integer, real number, etc.), rather than according to name.

The above example shows a list of objects on the left, and the available properties (e.g., bgcolor, height, width) of that object are displayed on the right. When a programmer is searching for a property of interest, the data type is meaningless, and only serves to make it more difficult to locate the desired property. One would have thought that they would have realized that this is a problem, given that this is the fifth major version of the product.

One of the features of *PowerBuilder* is that it allows the user to specify his or her own events, based on operating system messages. These messages are presented to the developer through a list, from which he or she selects the message of interest. The drawback is that there are hundreds of messages, each of which *PowerBuilder* unnecessarily attaches its own "pbm_" prefix to. Because of this prefix, the user cannot jump through the list by pressing the first character of the desired message, but instead, must physically scroll through the list using the scroll bar. The user ends up wasting a great deal of time searching through the list, when a few keystrokes could have made the process much more efficient.



Based on their decision to post an advertisement on their users' Start Menus, the Start Menu folder for *Datastream's Demo* would be more appropriately entitled, "The Leader in Arrogant Software".

The interface requirements of an installation program are no less important than the interface requirements of the software to be installed. The installation program for *Datastream's Demo*, an application that provides a preview of Datastream software, provides a number of examples of how not to design an installation program:

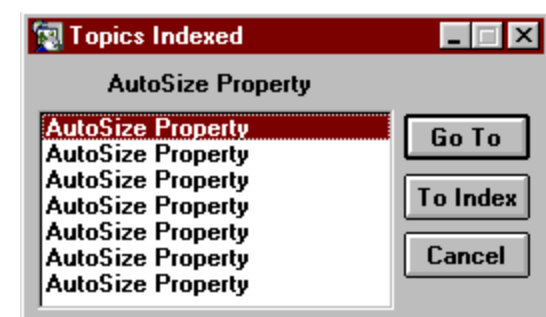
- Creates a folder that contains a single item, rather than simply adding the item.
- Does not allow the user to specify the folder into which the item should be installed.
- Uses an unnecessarily lengthy title that more than doubles the normal width of the start menu.
- Does not provide an uninstallation program, nor register itself with the Windows95 Remove Software applet. Removing the software will require an experienced user at least 10 minutes of effort, and even then, traces of the program will likely remain on the system.

The installation program provides your users the first impression of your software. How intelligent your software may seem will be determined in part by the apparent intelligence of the installation program. In this case, Datastream may want to reconsider how they showcase their software.

Stray from Microsoft's mainstream applications and you can easily find instances of interface stupidity. This example can be found in the Index function of the *Office Development Kit*. The user selects a topic of interest from the index, and is then presented a list, as shown here, of all instances of the topic in the files. The problem is, (and this shouldn't take a brain surgeon to figure out), is that the list provides no indication of the context in which the topic is discussed in each instance. The user must "Go To" each instance to determine which relates to his or her concern.

In this particular list, the first instance discusses AutoSize as it relates to pushbuttons, the second instance discusses AutoSize as it relates to picture controls, the third as it relates to label controls, and so on.

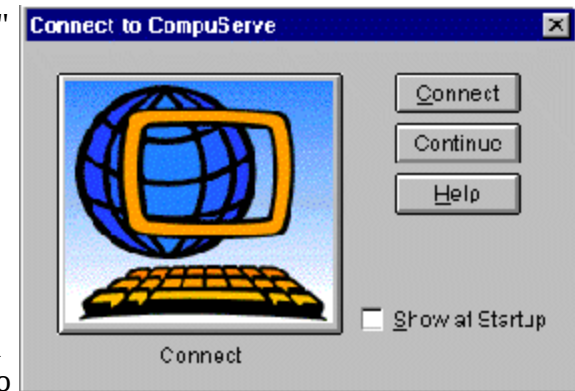
The program could easily determine the context of each instance, but for some reason, the programmers decided that they had better



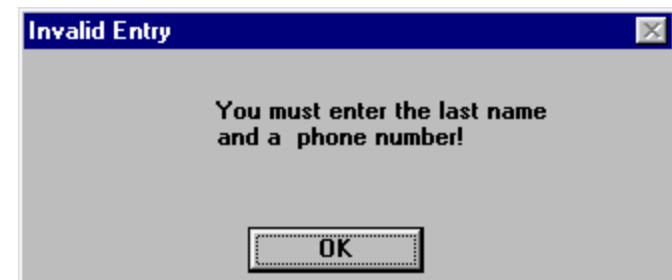
things to do.

Imagine the following scenario: You walk into a bar and say to the bartender, "I want a beer." The bartender responds, "Do you want a beer, or something else?" You think to yourself, "perhaps he didn't hear me", so you again say, "I want a beer". The bartender responds, "Do you want a beer, or something else?" You stand there perplexed, "is this guy an idiot, or am I living a Kafka story?"

Connecting to CompuServe through their *WinCim* application is not unlike this scenario. When you select **Connect** from the **File** menu, the application responds with the dialog box shown to the right. After the user has issued a command, the fact that *WinCim* asks, "Do you want to connect, or do something else?", makes it seem pretty darn stupid. You need to again state that you want to connect, or press the **Continue(?)** button to dismiss the dialog and ... do something else.



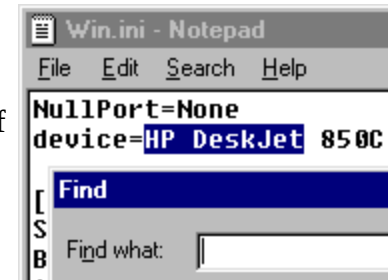
This attitude expressed in this error message from IBM's *Aptiva Address Book* application, certainly earns it a place in our [Error Messages](#) section, but it illustrates two aspects of interface stupidity. The first is that the message is wrong; it is generated when the user attempts to add a record that's missing *either* the last name or the phone number. The programmer apparently was not competent enough to display a message for the specific omission, so he or she opted for a generic incorrect message.



The second, more serious aspect of interface stupidity is that the program has adopted arbitrary rules based on the needs of the program rather than the needs of the user. The application requires the user to enter a last name and a phone number for each record. The end result of these rules is that they compromise the usefulness of the application. How does the user enter a record for the local power company? Do I have to know the name of the receptionist at the cable company just to store the number of the billing department? What if I want to enter the mailing address of a company and I don't have the phone number - after all, it's an **Address** book! The only way to do so is to enter a *fake* phone number.

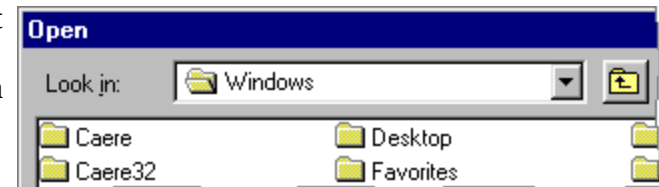
The problems illustrated with this application are unfortunately all too common, especially in corporate applications. They result when the application is designed according to the needs of the database, rather than designing the database according to the needs of the user. Any time you hear the phrase, "required field", an alarm should go off warning you to determine how it will limit your users.

This one is a personal pet peeve of the author. In the *Visual Basic* programming environment, if you select a piece of text then open the Find dialog, the highlighted text will be displayed as the text to search for. This is as it should be. Unfortunately, few other programs are smart enough to figure out that if you open the *Find* dialog immediately after selecting some text, you are likely to be searching for another instance of that same text.



As shown in the image, *Notepad* is unable to figure out that I might just want to search for "HP Deskjet". Similarly, Microsoft's *Word*, *Excel*, and *Access* programs are equally dumb. At least, the developers decided to make them that way. The developers of Microsoft's *Wordpad* on the other hand, decided that having to write an extra line or two of code was not too high a price to pay for making the user's life a little easier.

How many bitmap files do *you* keep in your Windows directory? Every time you start *Microsoft Paint* and attempt to open a file, the starting directory is always the Windows directory. For most users, this directory contains more folders and files than any other directory. As such, it is probably the last place you would look for a bitmap file.



Most users keep their bitmap and graphic files together in one of a few separate directories on their computers. A *smart* program would be able to remember the last directory selected by the user. Unfortunately, *Paint*'s developers took the easy way out, making your computer dumber in the process. Writing a smart program can sometimes require a little extra effort. In this case, it would have required *very little* extra effort.

While it is possible to specify the starting directory to be used, this is accomplished **not** from within *Paint*, but by going ... somewhere else: open Explorer, locate the shortcut to the MS Paint application, right-click on it, select Properties, select the ShortCut tab, and directly type in the desired directory. This is how we believe Microsoft came up with the "Where do you want to go?" ad campaign: it's a shortened version of the question, "where the heck do you have to go to change the default directory in *Paint*?"

Tooltips were intended to provide descriptive information to help new users learn the functions of *graphical* toolbar buttons. When they are used for standard command buttons, they invariably elicit the following response from users:

Duh!



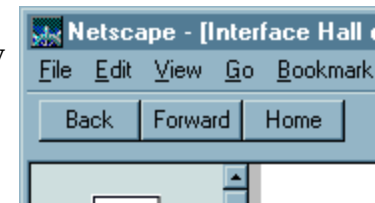
As shown in this image from *Mindspring's Pipeline+* internet access application, rather than providing useful information to the user, the tooltips merely convey that the designer is, well...let's just say, *intellectually challenged*.

Here's a rule that developers should keep in mind:

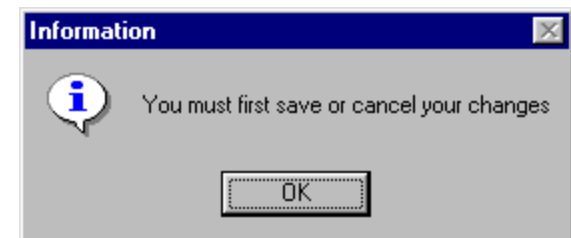
People generally don't like to use stupid applications



John wrote to describe this particularly helpful feature of *Netscape Navigator*, which can easily be replicated by specifying "Text-Only" toolbar buttons in the General Preferences section. It certainly got a chuckle out of us.

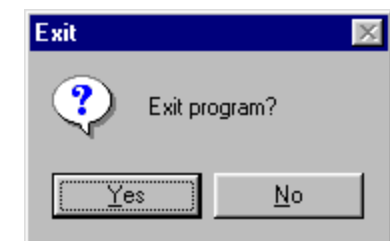


Attempting to exit *Visual Labels* can be a trying experience. When selecting the Exit command, the user is presented the following message, forcing the user to (1) respond to the message, and (2) perform one of the required functions.



Upon selecting Cancel, as directed by the previous error message, the user is then hit with the following message, sure to raise the blood pressure of any user.

This process could be handled much more smoothly, without requiring the user to jump through the programmer's hoops.



Microsoft's *Web Publishing Wizard*, is used to manage Web site files. At one point in the process, the Web Wizard obtains a list of the existing files on your



site then asks you, one file at a time, whether or not you want to delete the file. If we wanted to delete the file *zderr.gif*, for example, we would have to respond "No" to each of the nearly 600 filenames that precede it. Computers can't get much dumber than this.

If the program is smart enough to get the list of files, why can't it be smart enough to display the list, rather than showing them one at a time?

"Wizard" might be asking too much. Before connecting to the internet server, Microsoft's *Web Wizard* asks for the user's ID and password. The "Wizard" then starts the Windows95 dialer, but forgets to pass it the account information just entered.

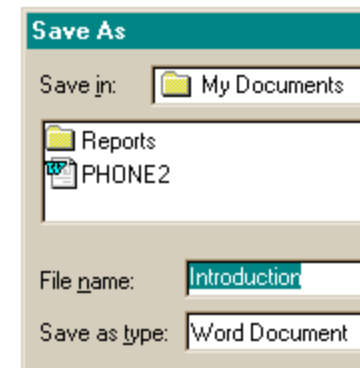


In a misguided attempt to add intelligence to *Office95* applications, Microsoft ended up demonstrating just how stupid a computer can be, and how infuriating it can be for the user.

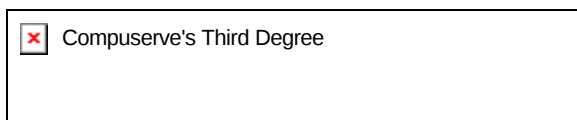
After selecting the Save function in an *Office95* application, the *Office95* **uncommon dialog** will be displayed, with a **suggested** filename, apparently taken from the first line of the document. The problem is clearly expressed in this note we received from **Chuck Layton**:



If you change the file name and then navigate to a different directory, the file name reverts to the suggested name. It drives me crazy! Am I supposed to say "Oh gosh! You're right! That file name makes a lot more sense now that I'm saving it to a different directory!"?



Searching for a phone number in CompuServe's *WinCim* application makes you feel like you're sitting under a glaring light in a smoke-filled room. The application displays form



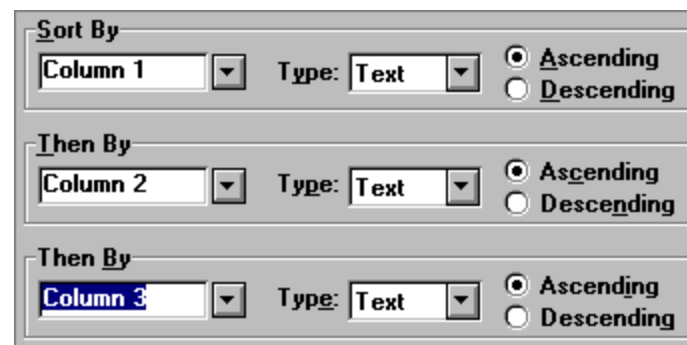
after form after form, each with a single choice to be made. In addition to those shown, there are two additional forms, but each appears maximized, so as to obscure the others.

Perhaps the designer thought that since the site is free, timely interaction with this function really wasn't a criterion. Then again, perhaps the designer didn't think.

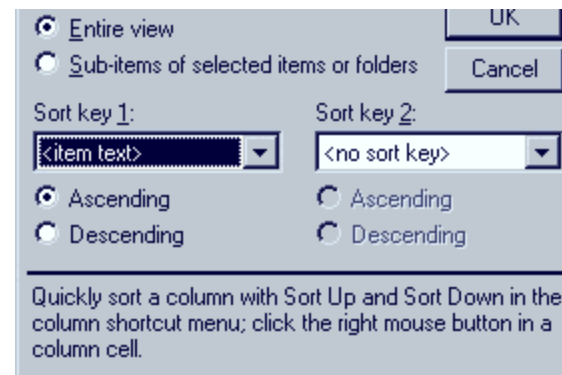
For some reason, the designers of *Microsoft Word for Windows 6.0* decided to limit the computer's intelligence. Your computer is certainly capable of sorting records on more than three fields, yet the designers chose to set an arbitrary limit, deciding that you *probably* wouldn't need more.

Because of this arbitrary limit, we cannot, for example, sort our clients by State, City, Last Name, and First Name. The designers figured we'd never need to.

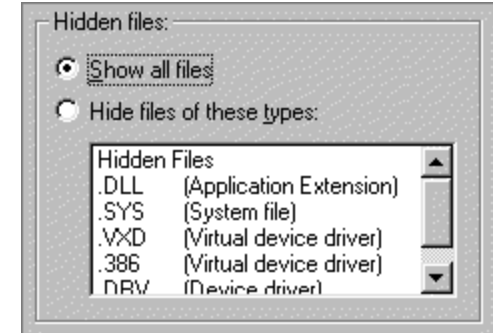
The old saying, "When you *assume*, you make an *ass* out of *u* and *me*", still holds true.



Netmanage Ecco Pro (v4.01) is purported to be a "very sophisticated personal information manager". Some magazine reviews praised its flexibility, while others complained about its rigid structure. The reviewers in the first category probably never tried the Sort function, which needlessly limits the user to only two sort fields. Want to see your contacts sorted by State, City, and Name? Ain't gonna happen. To Netmanage, we must ask, "Why Not?"



Win95's *Explorer* allows you to exclude certain file types from being displayed. That's nice. Unfortunately, they don't allow you to choose which types can be excluded; you either exclude no files, or you exclude all of the files in a set predefined by Microsoft. That's not nice. Developers for example, are often interested in .DLL files, but to have them displayed, they also have to display .DRV, .VXD, .SYS, and .386 files, all of which are meaningless to all but the most sophisticated propeller-heads.



[Home](#) - [Design](#) - [Announcements](#) - [Shame](#) - [Fame](#)

© 1996-2000 Isys Information Architects Inc. All rights reserved.
Reproduction in whole or in part in any form or medium without express written permission is prohibited.
bchayes@iarchitect.com