Why doesn't the mail image resizer check the image size before offering to resize?

devblogs.microsoft.com/oldnewthing/20091005-00

October 5, 2009

Raymond Chen

Commenter Igor lambastes the image resizer dialog that appears when you select Send To Mail Recipient. (And people think I'm the one with the social skills of a thermonuclear device.) This dialog pisses him off so much, he complained about it again.

The root of the diatribe appears to be that the image resizer dialog appears, even if it turns out the resizer won't do anything. For example, the resizer dialog appears even if the images are already small, or if the files have a .jpg extension but aren't actually JPG images, Why is it so idiotic that it fails to check these simple things before offering to do its work?

Because checking these simple things before showing the dialog is even more idiotic.

One of the grave errors when doing work with files is accessing the file before the user asks for it. This is a grave error because accessing the file can take an excruciatingly long time if the file is stored on a server halfway across the world over a slow network connection, or if the file has been archived to tape.

This particular code path is sensitive to the file access time because the user has just picked a menu item. Suppose the dialog box went ahead and opened the files to confirm that, yes, they really are images, and yes, the dimensions of the image are larger than what the dialog offers to resize them to. You select 1000 small images on a slow server, right-click them, and pick Send To... Mail Recipient.

Then you wait 30 minutes while the dialog box goes off and does something like this:

```
shouldOfferResize = false;
foreach (file in selection)
{
  if (file.IsJPGThatIsNotCorrupted() &&
      file.IsWorthResizing()) {
    shouldOfferResize = true;
    break; // can early-out the loop once we find something
  }
}
```

Opening each file, parsing it to verify that it is a valid JPG file that decodes without error, and extracting its dimensions takes, say, 2 seconds per file. (The file is slow to access, say, it's on a network server or on a slow medium like a CD-ROM or a tape drive. Or the file is large and it takes 2 seconds to read it off the disk and parse it to verify that there are no decoding errors.)

After about 15 seconds with no response, you give up and say "I hate computers." and go off and do something else, frustrated that you were unable to email your photos.

And then in the middle of working in your word processor, this dialog box suddenly appears: "Windows can resize the pictures you send in e-mail so that they transfer faster and are easier to view by the recipient."

Gee thanks, Windows, for finally getting around to asking me about that thing I wanted to do *a half hour ago*.

Idiot.

And then when you click No, Windows has to go and decode the files a second time in order to print them. (Unless Igor's recommendation is to cache the decoded bits from the first pass. Then you'd complain that selecting 1000 files and clicking "Send To… Mail Recipient" causes your computer to run out of memory. As Igor is fond of saying when insulting the Windows team: "Looks like this feature was designed without any adult supervision.")

Sidebar: A good fraction of these blog entries are just elaborations on very simple concepts. When I toss an entry onto the "I should blog this" pile, it usually gets a short remark of five to ten words which captures what I want to say. Then when it floats to the head of the queue, I expand those ten words into a 300-word entry. The short version of today's entry: "That would hit the disk."

Raymond Chen

Follow