Are you sure? Yes. Oops!

Marc Stiegler, Alan H. Karp

HP Laboratories
HPL-2011-96

Keyword(s):

Confirmation errors; usable security; interaction design

Abstract:

Confirmation dialog boxes are common. So common, in fact, that we have trained ourselves to click through them without thinking. Most of the time, that's not a problem, but in rare cases it can lead to the loss of work. In this report, we'll come up with a plausible explanation of why that happens and propose an approach that we believe will make such errors rarer.

External Posting Date: July 06, 2011 [Fulltext] Approved for External Publication
Internal Posting Date: July 06, 2011 [Fulltext]

Are you sure? Yes. Oops!

Marc Stiegler, Alan H. Karp

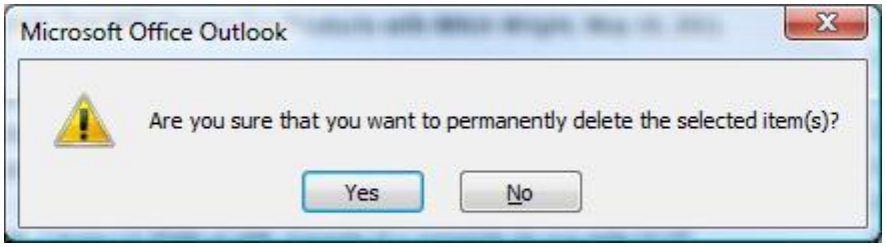
HP Labs

Abstract

Confirmation dialog boxes are common. So common, in fact, that we have trained ourselves to click through them without thinking. Most of the time, that's not a problem, but in rare cases it can lead to the loss of work. In this report, we'll come up with a plausible explanation of why that happens and propose an approach that we believe will make such errors rarer.

Introduction

We've all done it. You select some irreversible action, get presented with a dialog box asking you to confirm your decision, click OK, and only then realize that you've made a mistake. Figure 1 shows the dialog box you get when attempting to delete an item from Outlook's Deleted Items folder. Once you click Yes, it's too late. The item is gone.



Microsoft Office Outlook

Are you sure that you want to permanently delete the selected item(s)?

Figure 1. A confirmation dialog box.

Other examples include deleting an item from the Window's Recycle Bin and deleting a user account from the machine. Our favorite place where confirmation is not required but we think should be is on email Reply- All.

Of course, the best approach is to make all actions reversible. That's the point of having a Deleted Items folder in Outlook and a Recycle Bin in Windows. However, there are times where reversibility defeats the purpose of the action. Once a Reply-All email has been sent, it cannot be called back, and records retention policies often require complete (or as complete as is humanly possible) destruction of documents. In these irretrievable/not-undoable cases it's important to reduce the likelihood that the user will make a mistake.

Reducing Confirmation Errors

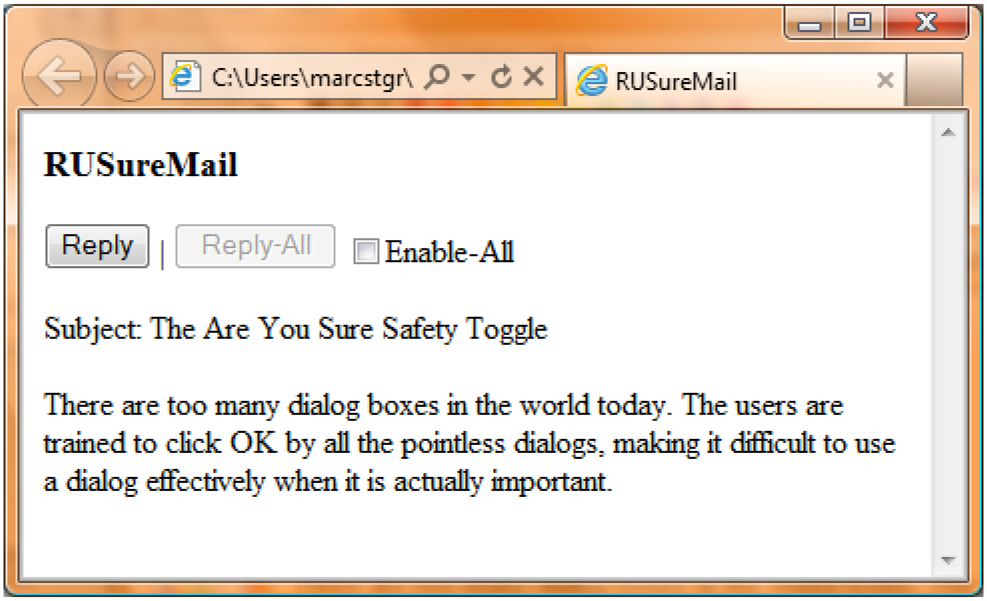
The standard approach is to ask for confirmation after the user has designated the desired, irreversible action, which we refer to as *post-confirmation*. The flow for post-confirmation is

1. Select the item to be acted on.
2. Select the action to be taken.
3. Confirm the action.

We propose using *pre-confirmation* instead. The idea is simple. For each irreversible action, there is a button or check box that enables it. The flow for pre-confirmation is

1. Select the item to be acted on.
2. Enable the action.
3. Select the action to be taken.

If the selected action wasn't enabled, the button or menu item for the action would be disabled and grayed out, as shown in Figure 2. Notice that pre-confirmation has the same number of steps and the same number of clicks as does post-confirmation. Unlike post-confirmation, where the user gets "trained" to click through the dialog box, pre-confirmation "trains" the user to enable the irreversible action first. The enablement checkbox is lightweight compared to the dialog.



RUSureMail

Subject: The Are You Sure Safety Toggle

There are too many dialog boxes in the world today. The users are trained to click OK by all the pointless dialogs, making it difficult to use a dialog effectively when it is actually important.

Figure 2. A mockup of preconfirmation in a prototypical email system.

The Reply-All button is grayed out until its corresponding Enable checkbox is selected. A tooltip might be used to explain the need to enable when the user hovers over the Reply-All button. After an enablement, after the Reply-All button is used once, the Enable checkbox is unchecked, and the Reply-All button is once again disabled.

A Plausible Explanation of OK-Dialog Errors

We believe there are three factors that contribute to confirmation errors. First, dialog boxes are all too common. Most of them are asking us to answer a question that we couldn't answer even if they gave us sufficient information, which they don't. The result is that we have trained ourselves to click through them as quickly as possible so we can get on with our work. Second, no harm is done most of the time we click through a dialog box. The system does what we expect, which reinforces the training of clicking through without thinking. That's particularly true of the confirmation dialog boxes. The user has only rarely made a mistake, so clicking through is almost always the right thing to do, anyway.

We believe that there is a third factor that contributes to the kind of confirmation error we're discussing here. By the time the user has taken the action, such as hitting the Delete key, the user's attention is shifting to the next task to be done. When the confirmation dialog box appears, it no longer appears as a part of getting the previous task done; rather, it is an impediment to getting to that next task. The reinforcement of the first two factors enhances the likelihood that the user will click through without paying attention. With pre-confirmation, the user is enabling the irreversible action in the course of performing the action. Thus, we posit that the user's attention is still on the action, rather than in the process of being redirected to the next task.

Conclusions

We have introduced the concept of pre-confirmation and provided a plausible explanation for why it might reduce the frequency of confirmation errors. User studies are needed to test our hypothesis. It is our hope that readers of this report will include pre-confirmation in their user interfaces and test our hypothesis as part of a larger study.

Acknowledgement: We'd like to thank April Mitchell for providing useful feedback.