

Confirmation

A technique for preventing unintended actions by requiring verification of the actions before they are performed.¹

Confirmation is a technique used for critical actions, inputs, or commands. It provides a means for verifying that an action or input is intentional and correct before it is performed. Confirmations are primarily used to prevent a class of errors called *slips*, which are unintended actions. Confirmations slow task performance, and should be reserved for use with critical or irreversible operations only. When the consequences of an action are not serious, or when actions are completely and easily reversible, confirmations are not needed. There are two basic confirmation techniques: dialog and two-step operation.²

Confirmation using a dialog involves establishing a verbal interaction with the person using the system. It is most commonly represented as a dialog box on a software display (e.g., “Are you sure you want to delete all files?”). In this method, dialog boxes directly ask the user if the action was intended and if they would like to proceed. Confirmations should be used sparingly, or people will become frustrated at the frequent interruption and then learn to ignore them. Dialog messages should be concise but detailed enough to accurately convey the implications of the action. The message should end with one question that is structured to be answered *Yes* or *No*, or with an action verb that conveys the action to be performed (the use of *OK* and *Cancel* should be avoided for confirmations). For less critical confirmations that act more as reminders, an option to disable the confirmation should be provided.

Confirmation using a two-step operation involves a preliminary step that must occur prior to the actual command or input. This is most often used with hardware controls, and is often referred to as an *arm/fire* operation—first you arm the component, and then you fire (execute) it. For example, a switch cover might have to be lifted in order to activate a switch, two people might have to turn two unique keys in order to launch a nuclear weapon, or a control handle in a spacecraft might have to be rotated and then pushed down in order to be activated. The purpose of the two-step operation is to prevent accidental activation of a critical control. If the operation works only when the two-step sequence has been completed, it is unlikely that the operation will occur accidentally. Two-step operations are commonly used for critical operations in aircraft, nuclear power plants, and other environments involving dangerous operations.

Use confirmations to minimize errors in the performance of critical or irreversible operations. Avoid overusing confirmations to ensure that they are unexpected and uncommon; otherwise, they may be ignored. Use a two-step operation for hardware confirmations, and a dialog box for software confirmations. Permit less critical confirmations to be disabled after an initial confirmation.

See also Constraint, Errors, Forgiveness, and Garbage In–Garbage Out.

¹ Also known as *verification principle* and *forcing function*.

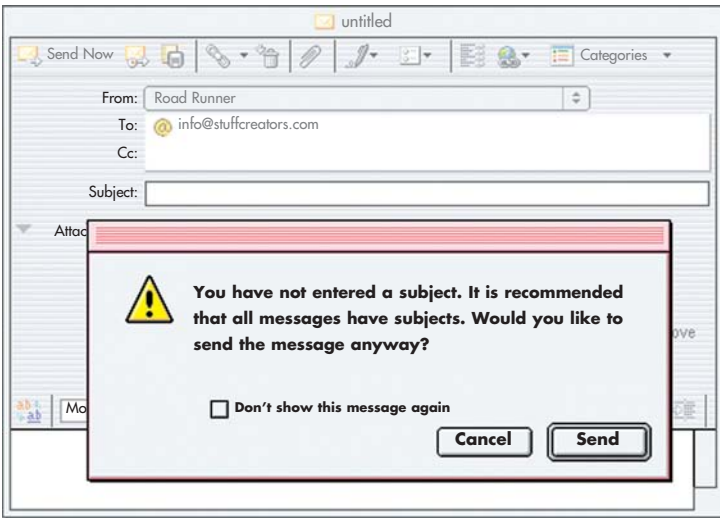
² See, for example, *The Design of Everyday Things* by Donald Norman, Doubleday, 1990; and *To Err Is Human: Building a Safer Health System* edited by Linda T. Kohn, Janet M. Corrigan, and Molla S. Donaldson, National Academy Press, 2000.

Register

Enter a UserID:

Enter a Password:

Re-Enter Your Password:



Launch Rocket

Common examples of confirmation strategies include: typing in a password twice to confirm spelling; confirming the intent of an action by clicking the action button (*Send*) with an option to disable future confirmations; having to remove a lock to open a valve; and having to turn two unique keys to complete a launch circuit.

