

## Re: Form Generation w/ tmp variables

---

*Source:* <http://www.tech-archive.net/Archive/Access/microsoft.public.access.forms/2008-03/msg00940.html>

---

- *From:* "Albert D. Kallal" <[PleaseNOOOsPAMmkallal@xxxxxxx](mailto:PleaseNOOOsPAMmkallal@xxxxxxx)>
  - *Date:* Fri, 21 Mar 2008 13:08:07 -0600
- 

"Mike P" <[MikeP@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:MikeP@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)> wrote in message  
[news:E5CA61A7-2908-47F8-9314-CEF47F64B6F2@xxxxxxxxxxxxxxxxxxxx](mailto:news:E5CA61A7-2908-47F8-9314-CEF47F64B6F2@xxxxxxxxxxxxxxxxxxxx)

I used the Create Form to generate a form for a main table. It works wonderfully; however, the users are working with the actual data. If they over type something in the wrong field, it is gone forever.

No, the above is incorrect. The record is not written back to the table until the user closes the form, or moves to another record (which by the way is NOT different then forcing them to hit save, and THEN move...

If the user made a mistake, have them go edit->undo.....

Is there a way  
to have the record displayed in the form be loaded with temp variables,  
then  
require the users to hit a Save button before the real data in the  
database  
is updated?

You can force the data to disk by going:

```
if me.Dirty = True then  
me.Dirty = false  
end if
```

However, you NOT explained to me how you going to deal with a person who to quote you:

If they  
over type something in the wrong field, it is gone forever.

What happens if they hit save? How is prompting the user to hit save going to solve this problem? Uses after a few days will hit the save button, and

## Re: Form Generation w/ tmp variables

ALWAYS answer yes without looking. So, they over type something, and hit save...answer yes....what have you accomplished?

So, just as a side note, the record in a form is NOT committed to disk until the user closes the form, or moves to another record. If the user overtyped a field, AND THEY DO NOT RECOGNIZE this fact, they STILL ARE going to hit save, and answer yes.

IF THE USER DOES recognize that they accidentally overtyped a field, just go edit->undo. And, if you go edit->undo a 2nd time, not only will the field be restored, the whole record of changes will be un-done (esc key does this also and is much faster then edit->undo).

So, adding a save button DOES NOT solve your problem here.

The ONLY thing you can hope here is that the user realizes they made a mistake, or have been editing the wrong record. If they recognize this fact, they simply just go edit->undo. (or hit Esc key, which means same thing)

If the user does not realize they made a mistake, why would they not answer yes to a save prompt???? this solves nothing.

The other problem is of course while you can force a save prompt on your users, it means you have to code this for every form, and worse they will now be nagged all day long.

This is much like putting a key in your car, turning the key, and your car then pops up a dialog box asking you do you really want to start the car? The fact of turning the key shows the user is ready to start the car. The fact that a user is moving to a new record means it quite obvious they want the current one saved.

So, the record is NOT committed to disk, and you can put tons of verification code in the before update event of that form, and if you set cancel = true, then the user will be prevented from writing the current record to disk (and, you can use me.undo in code to cancel the disk write \*if\* you want to dump the changes and not commit the record to the table). So, in effect, the current record is loaded into a buffer, and is already a copy of the original data.

The only real problem in access with bound forms is things fall down when you have sub-forms because when you move to a sub-form record the main form record \*is\* written to disk (this makes sense, since you can't add child records to a database unless the main parent record has been saved, and a primary key generated).

--

Albert D. Kallal (Access MVP)

Re: Form Generation w/ tmp variables

Re: Form Generation w/ tmp variables

Edmonton, Alberta Canada  
pleaseNOOSpamKallal@xxxxxxx