

## Re: Need technique to programatically update back end structure / relationships

---

*Source:*

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.tablesdbdesign/2005-07/msg00522.html>

---

- *From:* "Allen Browne" <[AllenBrowne@xxxxxxxxxxxxxxxx](mailto:AllenBrowne@xxxxxxxxxxxxxxxx)>
  - *Date:* Tue, 19 Jul 2005 10:10:12 +0800
- 

Hi Bob

Big question. Hopefully you will get some other responses to see what others do.

It's not too difficult to programmatically create new tables, fields, set properties, create indexes, and create relations using DAO. (DAO because it's the native Access library, and is the only way to set all the useful properties, bar a couple.) Post back if you need examples of how to do that.

So the next question is *\*where\** to run this code. One approach is to embed the code into the app itself. When it starts, it runs the normal startup code to test the back end is available, and then tests that one of the new tables is available. If not, it runs the code to create them. IMHO, it seems a bit top-heavy to permanently embed that kind of code into the app when it only needs to run once. (I also don't like the idea of code running each time the app starts that has the potential to modify the back end.)

An alternative approach is create a little mdb to perform the structural update. When "updater.mdb" runs, it asks the user to locate the back end for their app, and verify they have the right database by using the same uniquely named table you use for your reconnection-on-startup check. OpenDatabase() exclusively. Create each table, field, property, index, and relation, logging every step to a local table (in updater.mdb). If anything goes wrong, rollback and send yourself the logging table so you can see exactly where the process failed, and why. If it all succeeds, commit and set whatever property you use to handle versioning of the back end.

If you are comfortable programming in another language (such as pure VB), you can add a reference to the DAO library, and perform the same steps as described above from an executable rather than an Access MDB. This approach is suited to the runtime where Access may not be installed.

I suggest you don't perform the update with DDL query statements because they don't let you set important properties such as AllowZeroLength for text fields. Likewise, ADOX won't allow you to set a Format or Caption on the field – not a big deal, but it can be useful to set the Format of a field to

Re: Need technique to programatically update back end structure / relationships

percent or currency, and to set the DisplayControl for yes/no fields to check box.

--

Allen Browne – Microsoft MVP. Perth, Western Australia.

Tips for Access users – <http://allenbrowne.com/tips.html>

Reply to group, rather than allenbrowne at mvps dot org.

"Bob Howard" <Bob@xxxxxxxxxxxx> wrote in message  
[news:OJIVhg%23iFHA.3300@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:OJIVhg%23iFHA.3300@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

>

> My application is deployed as a front–end / back–end Access 2000

> application

> (on any Windows platforms using Win98 or higher) and is optionally

> packaged

> / distributed with Access 2000 Runtime (the CD version has the runtime

> library — the download version does not).

>

> This application is distributed to various clients and uses Inno Setup for

> installation of all software, documentation, shortcuts, etc. During the

> install, the user selects whether this is a full install (in which case an

> empty back–end database is restored in addition to the programs and

> shortcuts), or the user selects that this is a new version upgrade (in

> which

> case the back–end database is left alone and eveything else is replaced).

>

> During startup, the front–end MDE locates the back–end MDB and refreshes

> all

> the back–end links (if the location has changed since the prior

> execution).

>

> The next version of the application requires an update to the structure of

> the back–end by adding about 4 (maybe 5 — the design is not complete

> yet)

> new tables, adding relationships among these new tables (some will have

> referential integrity with cascading deletes, and some will have

> referential

> integrity and not cascade deletes), and adding a relationship between one

> of

> the new tables and one of the existing tables in the database (referential

> integrity with cascading deletes).

>

> When previous new versions were distributed, all that was required was to

> replace the front–end (as all I was releasing was new application

> functionality using the existing structure and data).

>

> But the next version will be different due to the additions to the

> back–end

> structure.

>

> Can someone point me to a decent technique that I might employ to update

Re: Need technique to programatically update back end structure / relationships

> the  
> back-end MDB's structure while preserving the client's data that's already  
> within that database?  
>  
> Thanks in advance! !!  
>  
> Bob (@Martureo.Org)

---

• *Follow-Ups:*

- ◆ *Re: Need technique to programatically update back end structure / relationships*  
    ◇ *From:* peregenem
- ◆ *Re: Need technique to programatically update back end structure / relationships*  
    ◇ *From:* Bob Howard

• *References:*

- ◆ *Need technique to programatically update back end structure / relationships*  
    ◇ *From:* Bob Howard

- Prev by Date: *Re: How to update only 1 table*
- Next by Date: *Re: Need technique to programatically update back end structure / relationships*
- Previous by thread: *Need technique to programatically update back end structure / relationships*
- Next by thread: *Re: Need technique to programatically update back end structure / relationships*
- Index(es):
  - ◆ *Date*
  - ◆ *Thread*