

# Securing your database – Version 1.3

I have written various articles elsewhere on this website about different ways of making your databases as secure as possible. For example, see: <http://www.mendipdatasystems.co.uk/improve-security/4594461803>

I thought it might be helpful to explain how many of these methods are done

## IMPORTANT:

Many changes will only take effect after closing and reopening your database.

This is important as it will allow you to apply several security features at the same time.

**If you follow all the steps described, you will also be unable to modify your own application**

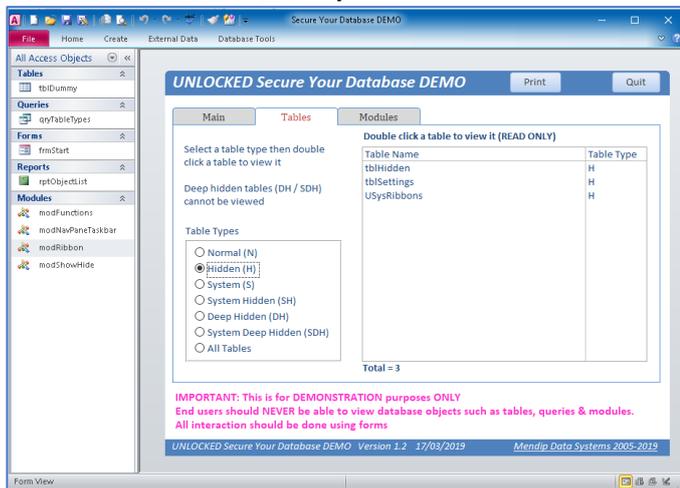
**ALWAYS ensure you have at least ONE copy of your file that is NOT locked down (preferably several copies)**

The example app attached to this article is initially **UNLOCKED** with full functionality available.

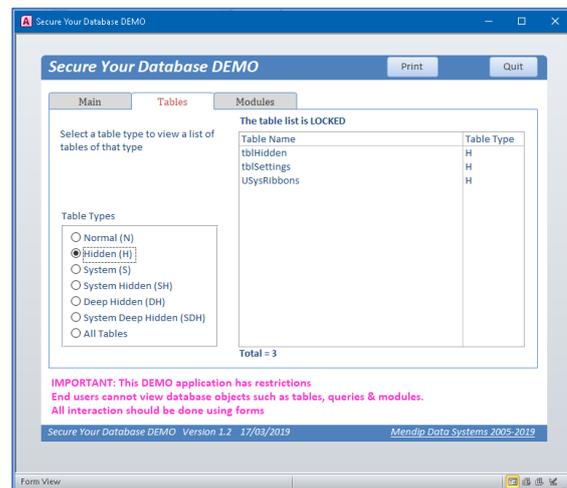
As you begin to lock this down, some functionality will be removed (either by that change or using code)

For example, you will no longer be able to view tables & module code from the form when the application is partly or fully **LOCKED** by hiding the navigation pane/ribbon/converting to ACCDE etc.

## UNLOCKED – Ribbon / Nav pane visible



## LOCKED - with form visible ONLY



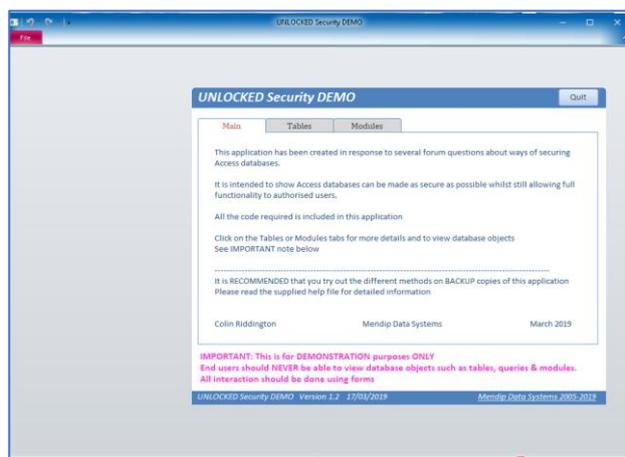
All code required make these changes is available in the supplied **DEMO** application

## NOTE:

All screenshots are from **Access 2010** unless stated otherwise. Some items may look different in other versions

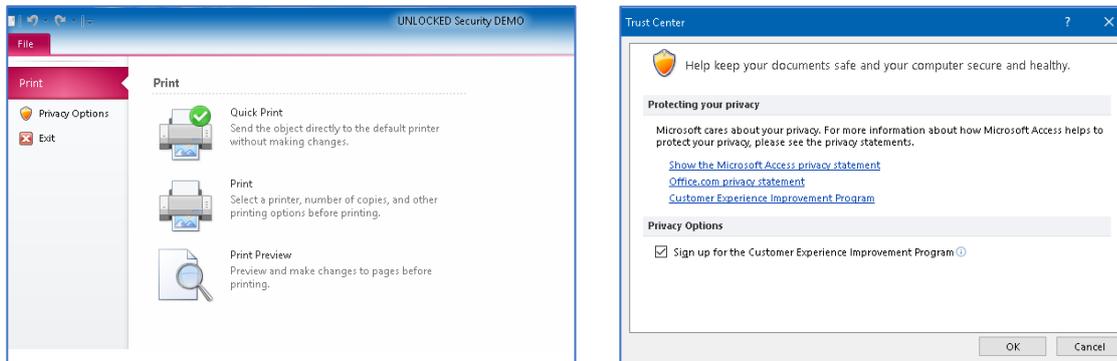
### 1. Use a Runtime version

If your users have a **runtime version** of Access, much of the functionality is disabled



In runtime mode, the **navigation pane** is removed and **right clicking** is disabled  
Users have no means of viewing/editing/deleting database objects.  
All interaction is therefore done using forms which is exactly what you want to achieve.

Furthermore, the **entire ribbon** is removed apart from the **File** menu and that only has limited options  
In this setup, **Privacy Options** does not open up any means by which users can obtain additional 'privileges'

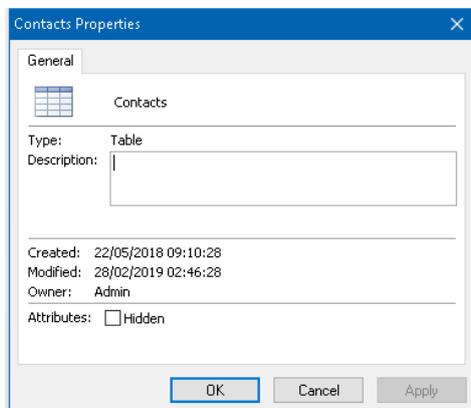


It is also possible to **emulate runtime mode** by changing the file suffix from **ACCDB** to **ACCDR**.  
Doing so will provide **sufficient security** to **prevent tampering** by **many users**.

However, users just need to **rename** the file with **ACCDB** suffix to regain full functionality  
For that reason, **runtime mode** is **NOT** a **reliable security measure** on its own.

## 2. **Hide all tables**

In the **navigation pane**, **right click** on a table then click **Table Properties**  
Tick **Hidden** in the properties dialog



This can be done for each table in turn or you can use the **HideAllTables** procedure in **modShowHide**  
Hidden tables will no longer appear in the navigation pane  
However, they can be made visible at any time if users click **Show Hidden Objects** in **Navigation Options**.

It is also possible to make tables '**deep hidden**' so they will **NEVER appear** in the navigation pane.  
Access also uses that feature for selected **system tables**

**However, for security reasons, I am deliberately NOT going to explain how to do that in this article**

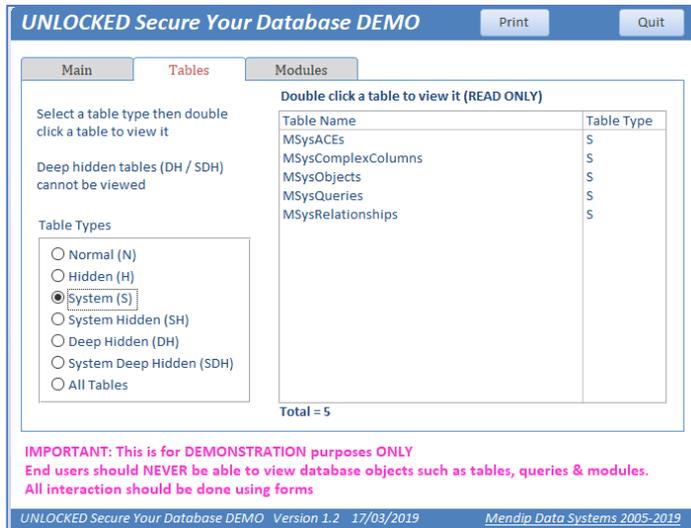
There is normally only 1 table (tblDummy) visible in the navigation pane.

However, there are, in fact, a total of 30 tables in this application including 3 hidden, 1 deep hidden and 25 system tables of different types

System tables can be made visible by ticking **Show System Objects** in **Navigation Options**. However, **deep hidden** tables are still not shown.

The **UNLOCKED** version supplied includes a feature allowing you to view the different types of table

Click the **Tables** tab then select a table type from the option group on the left. A list of tables appears. Click the **Print** button to view the table list as a report. Alternatively, double click a table in the list to view it (READ ONLY)



**NOTE:**

System tables are used by Access to make databases function correctly

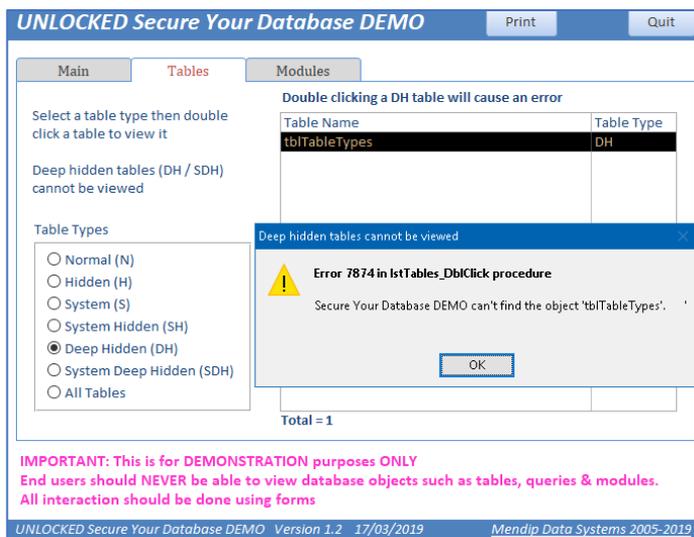
Some system tables can be viewed & a few can be edited

But that doesn't mean you should do so . . . UNLESS YOU ARE ABSOLUTELY SURE WHAT YOU ARE DOING

Altering one table may have 'knock on' effects on other tables

Incorrectly editing system tables may corrupt your database or prevent you opening it

If you click on any of the **deep hidden** tables, you will see a message similar to this



'Deep hidden' tables cannot be viewed by any standard method

### 3. Hide the navigation pane

A more useful approach is to hide the **navigation pane** completely  
Untick **Display Navigation Pane** in **Access Options** (see below)

Alternatively, use the **HideNavigationPane** function in **modNavPaneTaskbar**

A similar **ShowNavigationPane** function is also provided

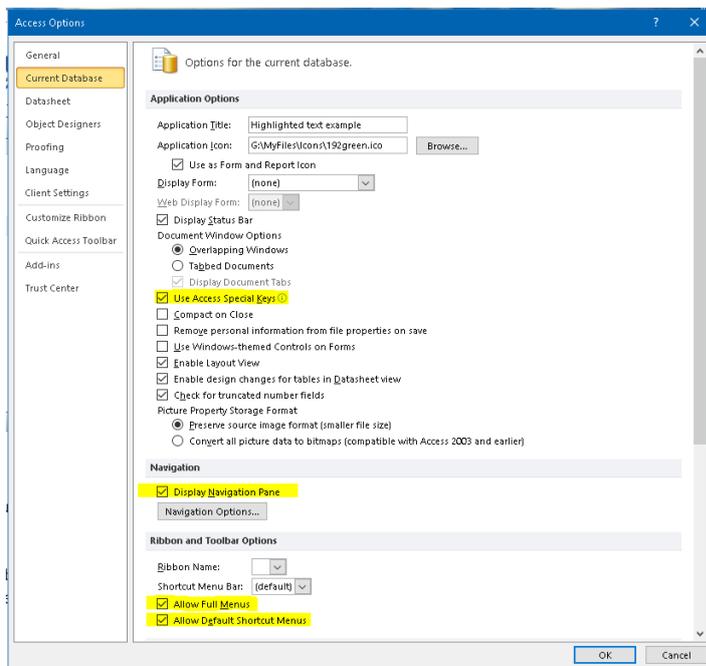
Both functions are accessible from the **Navigation Pane** or the **Modules** tab on the form

If you wish to remove the **navigation pane** 'permanently', do **ONE** of the following:

- Add the line **HideNavigationPane** to the **Form\_Load** event of the **startup form** in your application  
This code exists but has been disabled in my example app
- Create a startup macro called **Autoexec** and add the function to it using the **RunCode** command  
To see this in action, you can run or rename the inactive **Autoexec** macro I have created

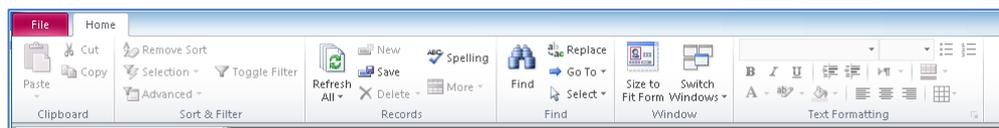
### 4. Disable various Access options

From the **File** menu, click **Options** then **Current Database**



**UNTICK** all the following items:

- **allow full menus** – this removes all menu items except **File** and **Home**

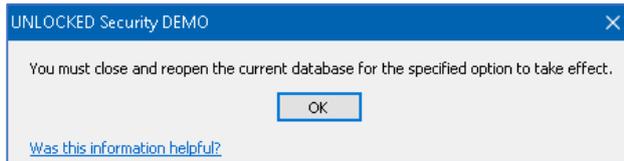


The **File** menu will look like the **Runtime** version with **Print / Privacy Options / Exit** items only  
However, clicking the **Privacy Options** item displays the full **Access Options** screen as shown above

- **allow default shortcut menus** – this disables right click context menus
- **use Access special keys** - this disables the following keyboard combinations that can be used in Access.
  - **F11** - shows & hides the navigation pane
  - **Ctrl+G** - Shows the Immediate window in the Visual Basic Editor
  - **Alt+F11** - starts the Visual Basic Editor
  - **Ctrl+Break** - prevents Access retrieving records from the server

**NOTE:**

After changing any of these options, you will see a message like this:



However, all these changes can still be reversed using **Access Options/Privacy Options** (see above)

**5. Remove ribbon menu items**

Do **ONE** of the following ..... BUT make another BACKUP first!

- Remove **Privacy Options** from the **File** menu (otherwise users can undo all the above)  
To do so, you need to create a new table called **USysRibbons** with 3 fields:  
ID (autonumber PK field), RibbonName (Text 255), RibbonXML (Memo/LongText)

In the **RibbonName** field enter **NoPrivacy** (or similar)

In **Ribbon XML** enter:

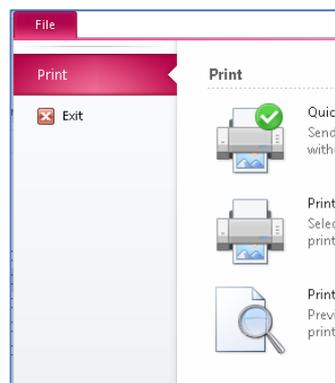
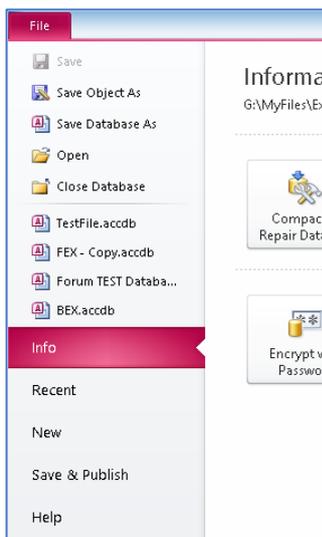
```
<customUI xmlns="http://schemas.microsoft.com/office/2009/07/customui">  
<ribbon startFromScratch="true">  
</ribbon>  
<backstage>  
<button idMso="ApplicationOptionsDialog" visible="false"/>  
</backstage>  
</customUI>
```

NOTE: A **USysRibbons** table with this code has already been created to save you time.

When you save the table, Access will treat it as a **system table** so it will not appear in the navigation pane  
Close and reopen the database. No change will yet be applied as the ribbon hasn't been selected.  
Go to **File . . . Current Database** and select **NoPrivacy** as the ribbon. Close and reopen again.

The menu now only has the **File** menu and **Privacy Options** have been removed

If you have already applied the previous changes, the **File** menu will be reduced to **Print** and **Exit**



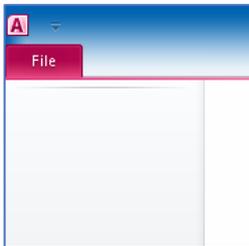
- Remove **all items** from the **File** menu  
Add a new record to **USysRibbons** and enter **SimpleFile** as the **RibbonName**  
In the **RibbonXML** field, enter

```
<customUI xmlns="http://schemas.microsoft.com/office/2009/07/customui">
  <ribbon startFromScratch="true">
<tabs>
  <tab idMso="TabHomeAccess" visible="false" />
</tabs>
</ribbon>
<backstage>
  <tab idMso="TabPrint" visible="false"/>
  <button idMso="ApplicationOptionsDialog" visible="false"/>
  <button idMso="FileExit" visible="false"/>
</backstage>
</customUI>
```

Change the selected ribbon to **SimpleFile** in **Access Options**

After you have applied this, close and reopen your application.

If you have also disabled items in point 3 above, you will now see an empty **File** menu:



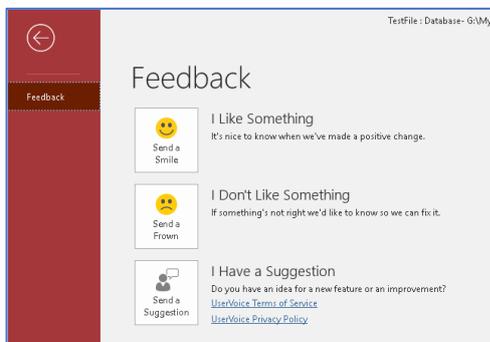
However, if you are doing this using a file with **PrivacyOptions** removed, you can't access this menu item!

The work-around if you are in this position is a bit convoluted:

- Rename your **USysRibbons** table temporarily to e.g. **XSysRibbons**.
- Close and reopen the database then **clear the ribbon name** from **Access Options**
- Rename the table back to **USysRibbons**. Close & reopen again
- Select **SimpleFile** as your ribbon in **Access Options**. Close and reopen again

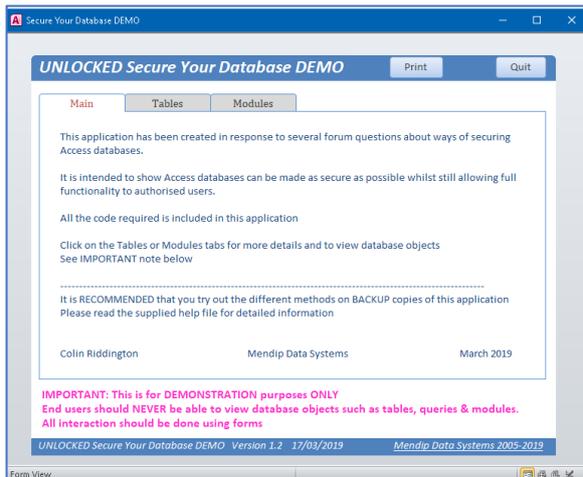
If you have already removed the navigation pane, go back to the backup you created earlier  
You did create that backup . . . didn't you?

NOTE: If you have **Access for Office 365**, the **Feedback** menu item will still be visible



I believe the **Feedback** item can be disabled but not removed completely  
However, I've not investigated this as it isn't a security risk

- **Hide the ribbon completely** using the **HideRibbon** procedure in **modRibbon**  
NOTE: this also hides the **quick access toolbar (QAT)**



A similar **ShowRibbon** procedure is also supplied  
NOTE: Do not use the **SimpleFile** approach as well as **HideRibbon**.  
The **USysRibbons** table actions will take priority

## 6. **Disable the Shift Bypass**

Users holding down the **Shift** key whilst **opening your database** can **bypass all start-up code** (and all security measures you've added). This is commonly referred to as the **Shift Bypass**

For example, using the **Shift Bypass**, users can display the **navigation pane** and **ribbon** even if previously hidden and use any Access special keys (and therefore have access to most/all database features)

**This is a serious security issue and none of the above security measures will prevent this.**  
However, the shift bypass can be **disabled** using **code**.

The module **modSecurity** contains a function called **DisableShiftBypass**.  
Before running this, you should make another BACKUP in case you lock yourself out!

You can run this from the **Form\_Load** event of your **startup form** (as for the **HideNavigationPane** function)  
This code exists but has been disabled in my example app

Alternatively, run it from an **Autoexec macro** (rename the inactive **Autoexec macro** in my example app)

### **IMPORTANT:**

The function works by first creating a property called **AllowByPassKey** (if it doesn't already exist) and then setting the property to **False**.

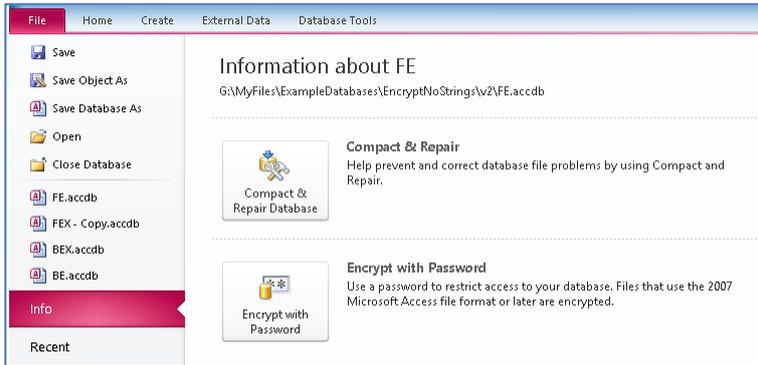
You will need to **close and reopen your database TWICE** to ensure the **shift bypass** has been **disabled**

### **NOTE:**

If you still have access to the **VBE**, you can reverse this by running the **EnableShiftBypass** function

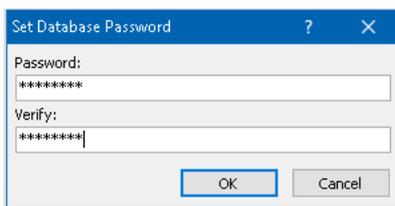
## 7. Encrypt using a strong password

- Open the database in **Exclusive Mode**. Click **File ... Info** then **Encrypt with Password**.

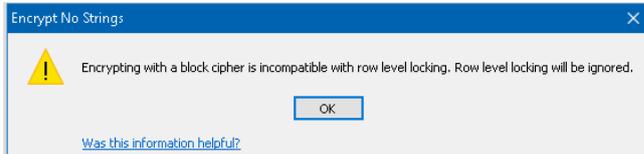


Enter a **strong password** that cannot easily be guessed (14 characters maximum) using a mixture of upper/lower case letters and numbers e.g. **DLt3Bgydi2tD** is good but **letmein** or **12345** are not.

Enter the password again to confirm

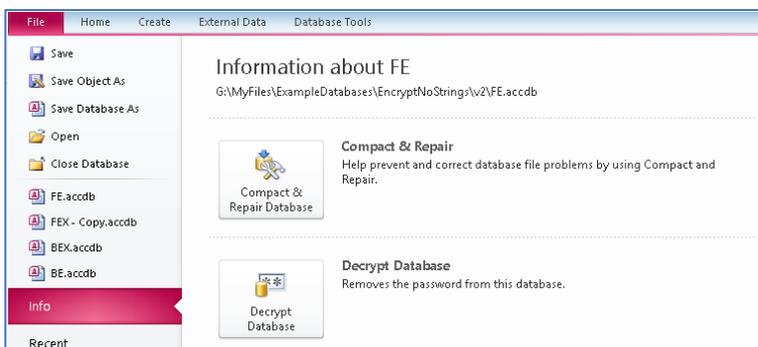


You will then see this message. Don't panic! Your database will be perfectly safe!



Close and reopen normally. You will need enter the password which is of course case sensitive

If you need to edit or remove the password later you will need to open it exclusively, enter the existing password then go to **File ...Info ...Decrypt Database** then make your changes.

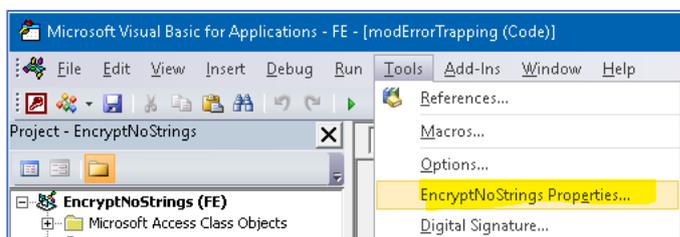


- Make sure the **password is memorable** but do NOT store it anywhere in the application (FE or BE) If you forget the password, you will be locked out!

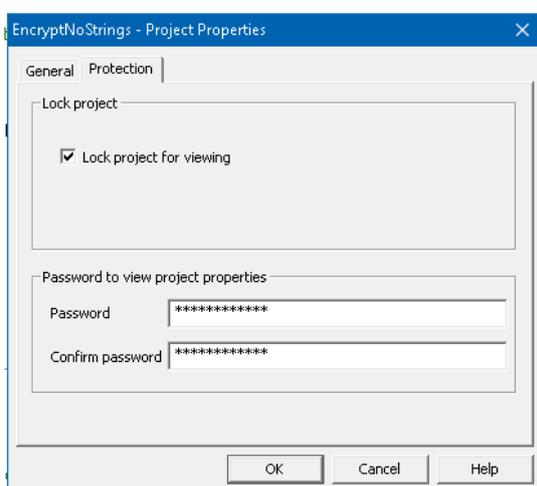
### HINT:

A strong password like that above can be based on a phrase that is memorable to you but not obvious to anyone else. See if you can work out the phrase on which **DLt3Bgydi2tD** is based...!

- Encrypting your database does more than just requiring users to enter a password. It also encrypts the application file so it cannot be read using a text or hex editor. See the article **Compare Access Security in ACCDB/MDB files**  
<http://www.mendipdatasystems.co.uk/compare-access-file-security/4594444323>
- It is also possible to just protect your **VBA** code with a password  
To do so, open the **Visual Basic Editor** on the **Database Tools** menu  
Then click **Tools ... (Database name) Properties**



This opens the **Project Properties** form. Now click the **Protection** tab  
Tick **Lock project for viewing**. Enter your chosen password twice and click OK



#### NOTE

**VBA project passwords** are MUCH less secure and a method exists which allow them to be bypassed.

Once again, for security reasons, I am NOT going to explain how this can be done.

**It is STRONGLY RECOMMENDED you do NOT rely on this approach alone**

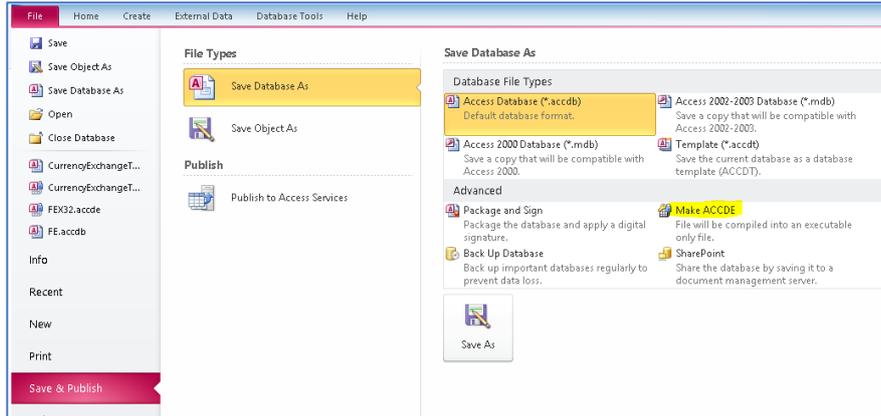
## 8. **Split your database**

Databases in a multi-user environment should **ALWAYS** be split into a frontend and backend application  
It is a good idea to also do this even if there will only ever be one user as it reduces the risk of data loss

- All tables should be moved to a shared backend database (BE) in a secure location on the server  
Users should connect to the **backend files** on a **LOCAL area network (LAN)**
  - RECOMMEND you do **NOT use a WIDE area network (WAN)** as it is likely to be slow and cause issues
  - **NEVER** use a **cloud based location** such as **OneDrive** or **Dropbox** as this **increases the risk of corruption** significantly
- Each user needs their **own copy** of the **front end (FE)** on their **own hard drive**
  - **NEVER** allow **multiple users** to run an **FE** from a **shared hard drive**
  - Users should **NEVER run a split database** when connected to the server using a **WIRELESS connection** as **corruption is almost inevitable** due to **dropped connections**

## 9. Convert the FE to an ACCDE file to make all code inaccessible

In the **File** menu, click **Save & Publish** then **Make ACCDE**



Access will only allow you to create an **ACCDE** file if the project is **fully compiled**

If there are any compilation errors, these will need to be fixed first

To do so, open the VB Editor then click **Debug...Compile**

Typically, compilation errors will be due to undeclared variables. Fix EACH error displayed in turn.

When the project is fully compiled, the **Compile** menu item will be disabled

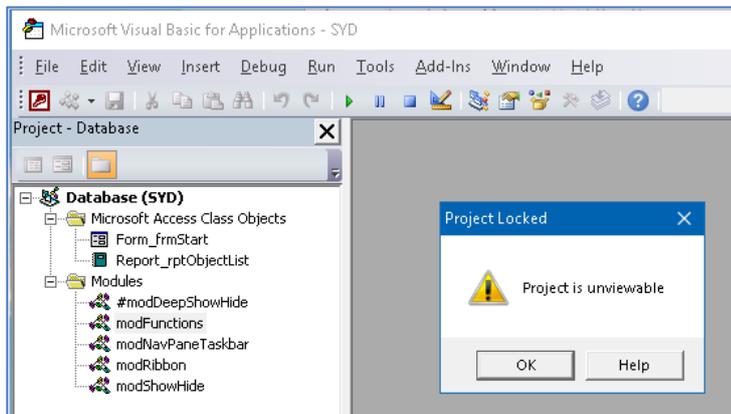
### NOTE:

Unlike **ACCDB** files, **ACCDE** files created in 32-bit Access will only run in that 'bitness'.

Similarly for **ACCDE** files created in 64-bit Access.

If you have some users in 32-bit & others in 64-bit Access, you will need to create **TWO ACCDE** files

Although users will still be able to open the VBE (unless prevented by other means), clicking on any database object will result in this message



There is no benefit in creating an ACCDE file for your backend database if it only contains tables

### IMPORTANT:

**Make sure you keep a copy of the original ACCDB file to allow continued development work**

It is **ALMOST** impossible to convert an **ACCDE** back to the original **ACCDB** file.

There are a few specialist firms that can **reverse engineer ACCDE** files but it will be very expensive.

Reputable firms will also require proof of ownership before proceeding

## 10. Apply security using database properties in code

It is possible to update almost all the above database properties using VBA.

The module **modSecurity** contains 3 other very useful procedures you can use:

- **StartUpProps / DeleteStartUpProps** – these contain the required functionality used in the procedure below
- **ModifyStartupProps** – this allows you to disable any or all of the following items using code
  - Allow Full Menus
  - Show Status bar at startup
  - Allow built in toolbars
  - Allow Shortcut menus
  - Allow toolbar changes
  - Allow special keys
  - Show database window at startup
  - Allow Shift bypass key

The procedure works by first deleting existing start up properties. It then reapplies them with all properties set to false

To test the effects of each, it is RECOMMENDED that you try each item in turn disabling all other items.

Once again, you may need to **close and reopen your database TWICE** to ensure the change is fully implemented

Once you have decided which items to use, you can add **ModifyStartupProps** to the **Form\_Load** event of the **startup** form or to the **Autoexec** macro

As this procedure duplicates many of the other methods described above, you may need to remove any repeated code

## 11. Encrypt Your Data

It is still possible for authorised users with advanced Access knowledge to gain access to your data from outside the application provided they know the encryption password.

**For security reasons, I am deliberately NOT going to explain how this is done**

However, it is also possible to **encrypt the data** itself.

Doing so means that anyone who does manage to retrieve your data tables will see something like this:

PersonID	Title	LastName	FirstName	Gender	DOB	Company	EMail
1	æ½	ø¢%7ZF)Vµ3T	ø»-↑~P	æ	»úcrFÖ↑~MÄ	äCE -T'G#ß,5  KI.	ø»--ZÏD.ËY τAA8'Ziu
2	æ½?	ä"®•	äi""↑E+	i	""úcrEÖ↑~NI	ñi*®  œß+ßS  LM ÖV yv	Ëx#-4Zi==S C ç>ÖH»s
3	æ½	ü\$%7I~L+	äa%~"~P	æ	»ýcsGÖ↑~B.Æ	Ü\$%/Tz ({~a6Q_2BP~ I-	Ä*;L\$YH1↑,
4	æ½	ñ-7"[4X"/]JM	äa®&	i	šücsEÖ↑~CÄ	æ#B"  œD	Ä*"®ZzZ;/ÆsD#O2Ñ
5	æ½	é½-S	é; /	æ	\$þcsCÖ↑~MÍ	ü ç#~M	É-†1+Ö {τ
6	í½	é#S  Z	á )	æ	""ýcsGÖ↑~LÍ	ä 6&↑J,	Äæ. L-Yü ↑,
7	æ  ?0	é\$#/'G#/'8	äa""+~"	i	šýcs@Ö↑~CÍ	i®.&-ZÄ&τ~¶A	Ä- " -1uH)↑VA<ÖUäfu3
8	ü#:	ø¢%7	ü +&-	æ	""+csGÖ↑~C.Æ	ä.~*	ü#;+ ↑J ;4~NCzK -ww
9	æ½?	æ >\$~L"	ø9]	i	»úcrFÖ↑~CÄ	äY ,_-šH)[ñ2' MO4YM«k	øçç)~ • \$§\$
10	æ½	ä <®	é .	æ	""úcsAÖ↑~NÍ	é .,~Lg1~"~] QçÖK%k	É\$#34½C ~""P\,Ziu
11	æ½	á \$-•%F)	é½%~+	æ	""ýcrEÖ~wKÇ	ä \$-•Gg ×~JJB.ÖW	ÉÉ&,"Z3 "" MD3IV ru6m";-Öi
12	æ½	üa""↑~G&	äE%5œ~>~"~\$	i	»úcsGÖ↑~CÄ	üa""↑~G&[E3~T11	Äx<L~"G&  /τV^<É)ç6y1-•?
13	æ½	ä®1.~"~M	Ü\$%/	æ	šæcsCÖ↑~LÄ		Ü%\$~ F~#~KB)ÜK  np 9
14	æc)	i®"S(Ö.~%>	i®""~(E&	i	""úcsMÖ↑~BÍ	ä \$-T~L0 ,	Iä ""~\$L5Z\$~F17B)\q p zCE'
15	æ½	æ~¿1+~%A	ü ñ1+œA	æ	""þcsEÖ↑~CÄ		Üc/\$~œ /;τ~K@~sBVE

For further information on this approach, see the article **Encrypted Split No Strings Database**  
<http://www.mendipdatasystems.co.uk/encrypted-split-no-strings-db/4594566347>

***Finally ....***

I hope this article has been useful to all Access developers

I would be grateful for any feedback on the article including details of any errors or omissions. If you do wish to respond, please email me at [info@mendipdatasystems.co.uk](mailto:info@mendipdatasystems.co.uk)

*Colin Riddington*

*Mendip Data Systems*

*Last updated 21/03/2019*