



Using Microsoft Access

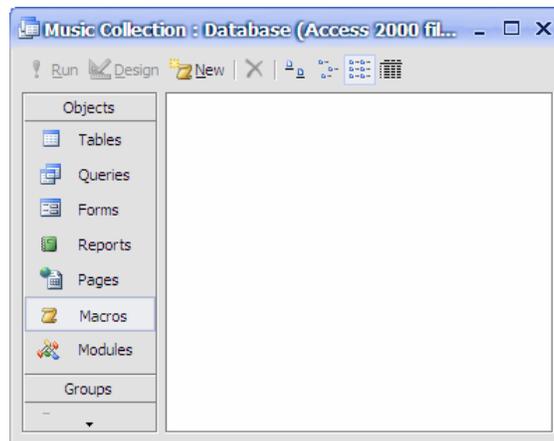
In the following exercises we will use a file called *Music Collection 3.mdb*. This database is similar to the one you worked on in the previous section except it has a lot of additional queries, forms and reports. Before you begin, you may want to open that database and familiarise yourself with the additional objects that have been created.

Basic Macros

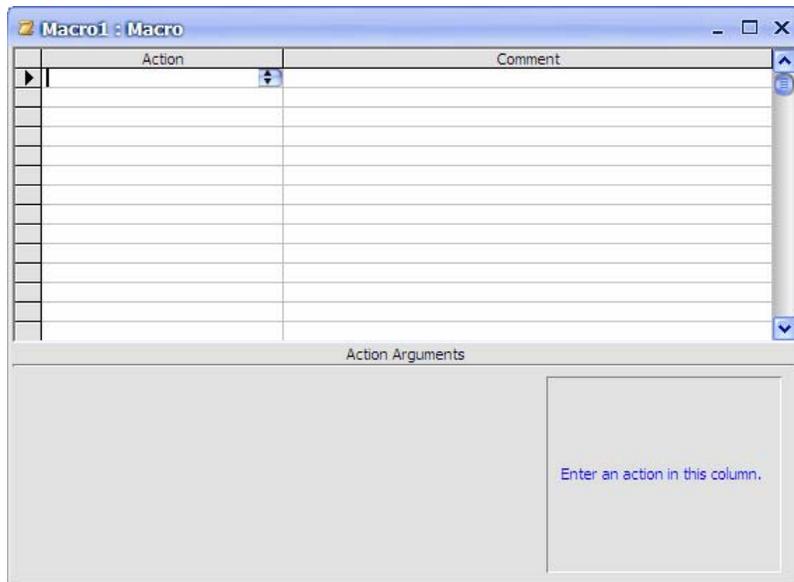
Macros provide a way to automate tasks that would usually be complex. Macros can be created easily in Access by choosing from a number of common procedures and combining them in whatever way you need. These macros can then be activated by various **events**, such as clicking a button or opening a form. Several similar macros can be grouped together to form a Macro Library. This keeps things better organised and makes it easier to locate a macro when you need it.

Exercise 1. Creating a Test Macro

- 1) Make sure the *Music Collection 3.mdb* database is open.
- 2) From the **Database Window**, go to the **Macros** section.

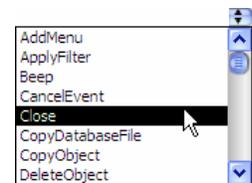


- 3) Click on the  **New** button to create a new macro.



You will notice that the Macro Design window looks a lot like the Table Design window. There is a top section for the steps in your macro and a bottom section for properties. We'll start by creating a simple macro that will close a window.

- 4) You may notice that the **Action** column has a drop-down list. Use that list to select **Close**.



Properties for this type of macro will now appear at the bottom. You can use these properties to specify what will be closed. For example, you could create a macro that will close a specific form in your database when it is used. If no properties are changed then it will just close whatever window is currently being used.

- 5) Click the **Save** icon  on the toolbar to save the Macro so we can test it.
- 6) Enter *Close Window* as the name for the macro.
- 7) Click the **Run** icon  on the toolbar to test the macro.

When the macro runs, it will close the current window (in this case, the macro design window). Later we will see how you can run a macro from other locations in the database.

Exercise 2. Creating a Macro with Multiple Steps

Now we'll create a macro that does more than one thing in a sequence.

- 1) Click on the  **New** button to create a new macro.
- 2) In the **Action** column, select *OpenTable*.
- 3) In the **Comment** column enter *Opens the CDS table*.
- 4) Change the properties at the bottom as shown (you can use the drop-down list to select from table names).

Table Name	CDS
View	Datasheet
Data Mode	Edit

- 5) Time to add another step. Return to the top section of the window and add a step under the first step as shown.

Action	
OpenTable	Opens the CDS table
OpenTable	Opens the SONGS table

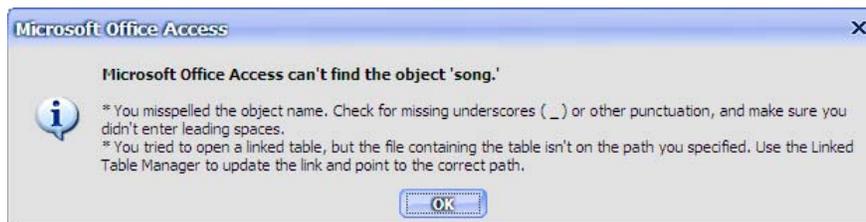
Table Name	SONGS
View	Datasheet
Data Mode	Edit

- 6) Save the Macro as *Open Main Tables* and run it. The macro will open both of the tables specified.

You can add as many steps as you need in a macro and each one will run in sequence until the last step is reached.

- 7) Close the tables and then close the **Macro Design** window.

Note Sometimes there may be a problem with your macro. This may be because it refers to an object that can't be found, often because it has been renamed or deleted. When a Macro can't run correctly, you may get messages like the following.



Exercise 3. Creating a Macro Library

A Macro library can be used to group several related macros together. In a large database with a lot of macros, libraries can be a useful way to organise them.

- 1) Click on the  **New** button to create a new macro.
- 2) Click the **Macro Names**  icon on the toolbar. A **Macro Name** column will appear to the left of the **Action** column. This column is used to provide a name for each macro within the library.
- 3) Enter some macros as shown below. Remember you can use the drop-down list to select Actions. There's no need to change properties for any of them.

Macro Name	Action	
Maximise	Maximize	Maximises a window
Minimise	Minimize	Minimizes a window
Restore	Restore	Restores a window

Each entry in the **Macro Name** column tells Access that what follows is a new macro, even if there is more than one step. Having a blank line between each one is not essential, but helps to make it easier to maintain. Later we will see how to refer to these macros and use them in the database.

- 4) Save the Macro library as *Windows Functions Library* and close the **Macro Design** window.

Note If you want to create a macro that runs in different ways depending on certain conditions, you can click the **Conditions** icon  on the toolbar while you are in **Macro Design**. This displays an additional column where you can specify conditions, and then the actions to take for each condition. It is similar to using an **IF** function in an excel spreadsheet.

Exercise 4. Using Macro Actions

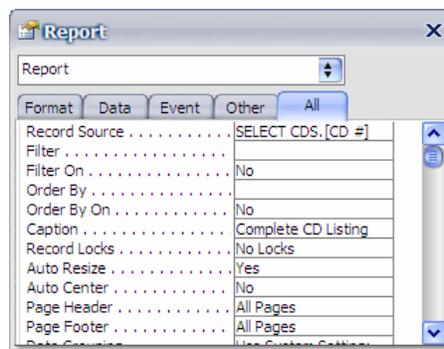
In the following exercise, we're going to make use of the macros we placed in the macro library. Our database has several reports which all display in print preview when opened. These reports are usually easier to preview when they are maximised (filling the screen). The problem is that if we maximise one of them then every other window in Access will become maximised but we have some forms that are better viewed in their normal restored state. We can make use of the *Windows Functions* macros to control whether a particular object opens maximised or restored. These Macros will be used in each object by the object's **event** properties.

- 1) Go to the **Reports** section of the **Database Window** and open the *Complete CD Listing* report (or another report that you have created if you don't have the *Music Collection 3* database).



When a report opens in a small window it will either cut some of it off or shrink it (like the example above) so that it is hard to see. We'll change the report's properties so that when the report opens, it will use the Maximise macro we created as part of our macro library.

- 2) Click the **View** icon  on the toolbar to enter **Design View**.
- 3) Once you are in design view, click the **Properties** icon  to view the properties window.



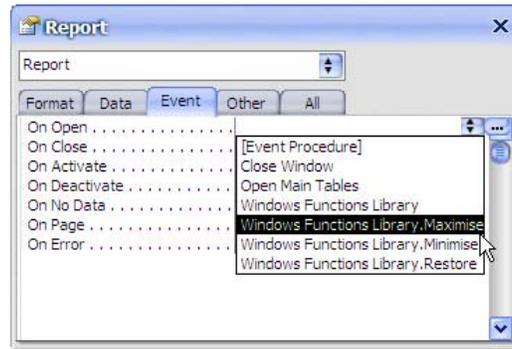
Every object in a form or a report has properties which can be changed in the properties window. The properties that are shown will depend on what part of the report is selected.

- 4) Make sure that the word *Report* appears at the top of the properties window as shown above. If it says something else instead then it is because you have a different part of the report selected. If something else is selected then click the square in the top left corner of the report (shown to the right) to make sure the properties for the whole report are showing.



The properties window has five tabs along the top. The first four group the properties in to categories. The last one places all of the properties for the selected object (the report in this case) in one long list. The properties we need for using macros are all in the **Event** tab. They are also found at the very bottom of the **All** tab.

- 5) Select the **Event** tab or the **All** tab and then find the **On Open** property.
- 6) Click in this property and you will notice that a drop-down list is available. Display this list to see all of the available macros.



The first macros we created are listed on their own. The ones in the Windows Functions Library are displayed with the name of the library and the name of the macro separated by a full stop. You may need to resize the properties window to make it wide enough to display the full macro names.

- 7) Select the *Windows Functions Library.Maximise* Macro from the list. This will cause the report to maximise as it opens. We will also make it restore again as it closes.
- 8) In the **On Close** property select the *Windows Functions Library.Restore* macro.
- 9) Close the report and save the changes. Open the report and close it again to test the macros.
- 10) Try applying the same macros to the other reports in the database.

Expressions

Anyone familiar with Excel will find expressions easy. An expression is a statement (usually a calculation) that produces a certain result. It is very similar to an excel formula and is written in much the same way. Like a formula in Excel, expressions in Access use the following rules:



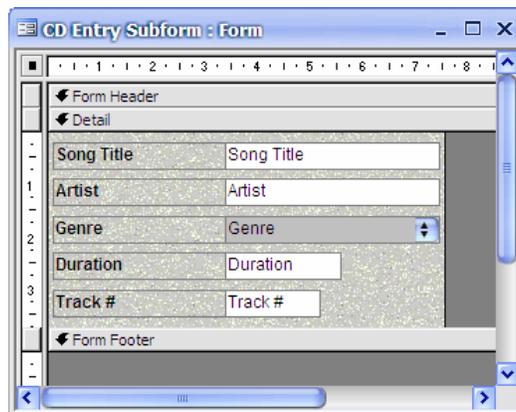
- An expression must begin with a = sign.
- An expression can use built-in functions (such as Sum) to simplify some types of expressions.
- Any calculations in expressions follow the order of operations, or the BIMDAS rule where different parts of the calculation are performed in the following order.
 - 1) **B** Anything in **br**ackets is calculated first.
 - 2) **I** Any **i**ndices are calculated next (E.g. =2^5 or 2 to the power of 5).
 - 3) **MD** Any **m**ultiplication or **d**ivision goes next.
 - 4) **AS** **A**ddition and **s**ubtraction are calculated last.
- Instead of cell references as in Excel, an expression will usually refer to objects such as text boxes. Any object names in an expression must be enclosed in square brackets.

You will notice that these rules are very similar to the rules explained in the advanced queries section for calculated fields. In some of the previous sections you already created some simple expressions to calculate totals on reports. Now it's time for some more involved expressions.

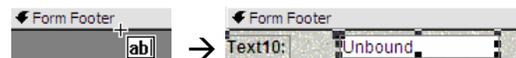
Because so many expressions rely on references to objects in your forms and reports, it is a good idea to provide meaningful names for the objects in your forms and reports. Since all of the forms and reports we have created use text boxes that are based on fields in your tables, most of them will have the same name as the field they are based on. A common practice among developers and programmers is to specify the type of object in the name of the object. For example, a text box that has been created to display a sales total might be named *txtSalesTotal*. This name would make it clear that the object is a textbox and would also clarify the purpose of the textbox. In a moment you will see how object names are used.

Exercise 5. Creating an Expression in a SubForm

- 1) Open the *CD Entry Subform* form in **Design View**.



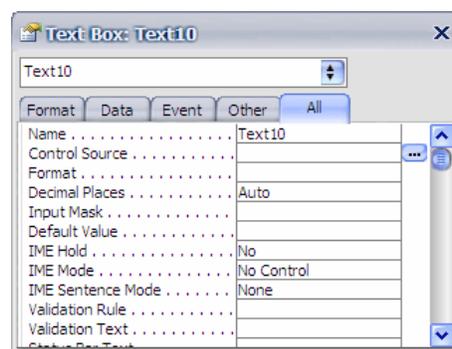
- 2) Make sure the **Toolbox** toolbar is displayed (usually on the left of the window) and click on the **Text Box** icon.
- 3) Click in the **Form Footer** to place a new text box there.



This textbox will be used to add up the number of songs in the subform. The result won't be visible in the main form but later we will create a text box on the main form that will refer to this one. Since the total on this subform won't be visible there is no need for the label next to the text box. You can delete it if you like. Make sure the new text box is still selected before proceeding.

Previously we have entered expressions by typing them in to the text box directly. This time we will do it using the properties window since we will change other properties as well.

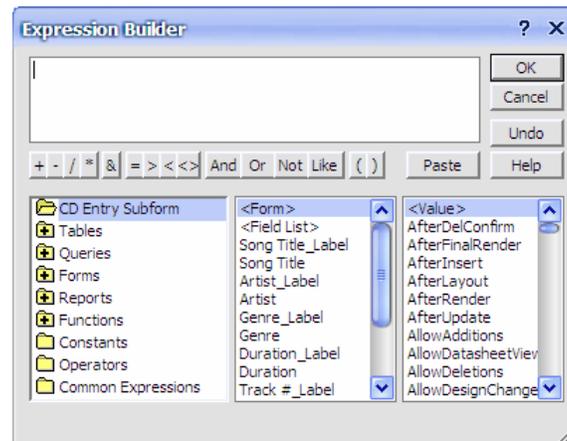
- 4) If the properties window is not already showing, click the icon on the toolbar to display it. If the text box is still selected, the properties window should appear as below.



- 5) In the **Name** property enter *txtTotalSongs*. This name will be used later when we refer to this object.

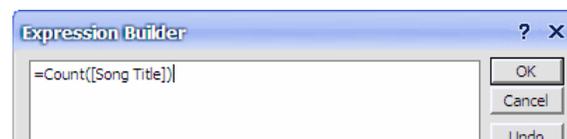
Note What you call the textbox is not important. What is important is that when you refer to it as we will do later, the name has to be typed the same including spacing (though it's not case sensitive).

- 6) Click in the **Control Source** property. This is the property which determines what will appear in the textbox so it is where our expression will go. You can use the right-click and **zoom** option shown in the queries section to get more room for editing but an even better way is to use the **Expression Builder**. This handy tool gives you quick access to common functions as well as making it easy to refer to database objects in an expression.
- 7) You will see an icon  to the right of the **Control Source** property. Click on this icon to open the **Expression Builder**.

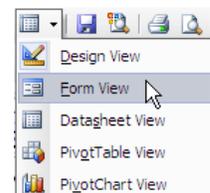


You can use the expression builder by typing directly in to the space at the top, or by using the icons and lists provided to insert segments in to your expression. We'll use a combination of both methods.

- 8) Start your expression by entering `=Count(` in the top section.
- 9) The middle of the three lists in the expression builder lists all of the objects in the form. Double-click on *Song Title* to add the name of that object to the expression.
- 10) Finish the expression by adding a closing bracket. The completed expression should look like the one below.



- 11) Click **OK** to complete the expression and return to the **Properties** window.
- 12) Click in another property to confirm the changes. This will avoid the occasional problem of your property changes not being kept.
- 13) Previewing the form normally won't show the total since this form is set to display in datasheet view by default. To view the form normally, click on the arrow next to the **View** icon and then select **Form View**. The total number of songs will appear at the bottom of the form.
- 14) Close the form and save the changes.

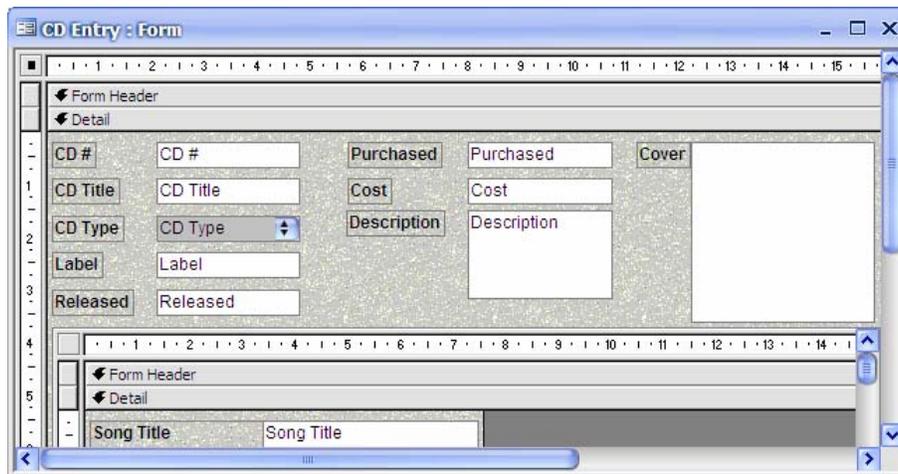


Exercise 6. Creating a Linked Expression

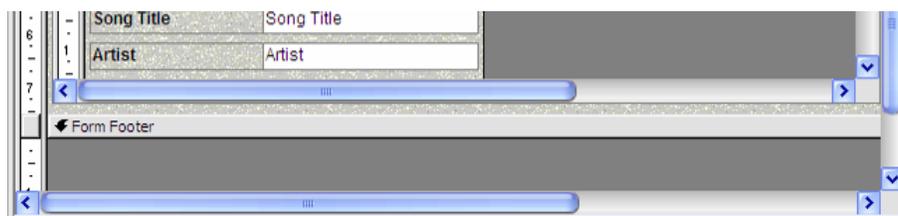
Now that our subform has an expression which adds up all of the songs, we will place an expression on the main form which adds refers to that subform total. Because of the relationships between our tables, the main form will only show the total of the songs showing at the time (i.e. the songs for each CD).

The tricky part here is that the expression on the main form needs to refer to the subform as well as the object within that form. Most mistakes in this kind of expression are made when the object name you are trying to refer to is typed differently when you refer to it.

- 1) Open the *CD Entry form* in **Design View**.



- 2) Scroll down until you can see the bottom of the form. We will place a text box for the total in the form footer.



- 3) Click the **Text Box** icon  in the **Toolbox** toolbar.
- 4) Click in the Footer area approximately 6cm from the left of the form to place a textbox. The ruler at the top should show cm measurements.

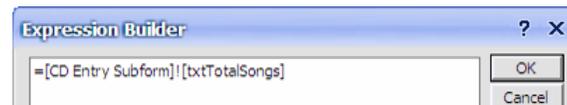


- 5) Make sure the textbox is still selected and open the properties window if it is not already open.
- 6) In the **Name** property enter *txtTotalCDSongs*.
- 7) Click in the **Control Source** property and then click the  button to open the **Expression Builder**.
- 8) Start the expression by typing an = sign.

The bottom-left list in the expression builder shows all of the objects in the database that can be used in expressions.

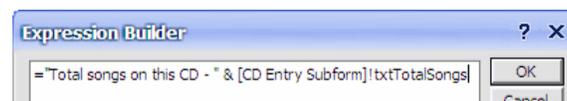
- 9) Double-click on **Forms** in the list to expand the selection.  Forms
- 10) Double-click on **All Forms** in the list to expand the selection.  All Forms
- 11) Double-click on **CD Entry Subform** in the list. The middle list will now display all of the objects from that subform.
- 12) In the middle list, double-click on txtTotalSongs. This will put a reference to that text box in your expression.
- 13) Remove the *Forms!* section from your expression. The completed expression should appear similar to the one below.

Track #_Label
Track #
txtTotalSongs
Form Header



- 14) Click the **OK** button to close the expression builder.
- 15) Click the **View** button  to preview the form. Your CD total should appear below the subform. View some different CDs to see the total change with each one.
- 16) Click the **View**  button to return to **Design View**. Now we'll get fancy by adding some text to the expression.
- 17) Edit the expression so that it appears as follows.

= "Total songs on this CD - " & [CD Entry Subform]!txtTotalSongs



The & symbol can be used to join text to a number in an expression.

- 18) Since we will have some text in the expression itself, there will be no need for the label. Click on the label next to the textbox and press the **[Delete]** key to remove it.
- 19) Resize the textbox so that it is approximately 6cm wide (you can use the properties for the textbox to make it exact by simply entering 6 in the **Width** property).
- 20) Preview the form to see the results of the expression and return to Design view once you are sure it is working properly.



Additional Form Formatting

Exercise 7. Formatting a Text Box

Now that our expression is working properly we can work on formatting. At the moment it looks like just another text box on the form. Since text boxes are often used for data entry, this may confuse some users. We can change to properties to that it looks more like a message that is integrated in to the form.

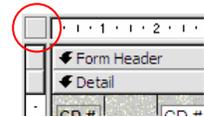
- 1) In design view, click on the *txtTotalCDSongs* text box.
- 2) Click the **Center** align icon on the toolbar. 
- 3) Click on the arrow next to the **Fill/Back Color** icon. 
- 4) When the colour palette appears, click on the **Transparent** option.
- 5) Click on the arrow next to the **Font/Fore Color** icon. 
- 6) Choose **red** for the font colour.
- 7) Click on the arrow next to the **Line/Border Color** icon. 
- 8) When the colour palette appears, click on the **Transparent** option.
- 9) Click the **Bold** formatting icon. **B**
- 10) Make sure the properties window is still displayed and change the following properties for that text box. These property changes will make the text box act less like a text box and more like a message.

Property	Change to:	Explanation
Enabled	No	User can't interact with the textbox.
Locked	Yes	Contents of the textbox can't be edited by the user.
Tab Stop	No	When the user is pressing the [Tab] key to move through the form it will skip this object.

- 11) Preview the form to see the changes and then return to **Design View**.

Exercise 8. Changing Form and Subform Properties

- 1) Double-click the top-left corner of the form (as shown to the right). The properties for the form should be displayed.
- 2) Change the properties for the form as shown in the table below.



Property	Change to:	Explanation
Caption	Add / Edit CDs	Changes the text which appears at the top of the form.
Scroll Bars	Neither	Any scrollbars on the form will be removed.
Record Selectors	No	The Bar on the left of the form that can be used for selecting a record will be removed since the form will only show one record at a time.
Dividing Lines	No	No visible lines between the form header/footer and detail.

- 3) Double-click the top left corner of the subform to view the properties for the subform.
- 4) Change the properties for the subform as shown in the table below.



Property	Change to:	Explanation
Default View	Datasheet	This property will already be set to datasheet but this is the reason why the subform displays as a datasheet table rather than a regular form.
Scroll Bars	Vertical Only	If you resized the columns in the subform so that they all fit there will be no need for a horizontal scrollbar.
Navigation Buttons	No	This removes the navigation buttons (next/previous record) from the subform. They are not really needed for the subform since it is easy enough to scroll through the records. Also, having navigation buttons for both the form and subform can be confusing for users.

- 5) Click the **View** button  to preview the form.

- 6) Save and close the form when done.

Exercise 9. Adding Form Navigation Tools

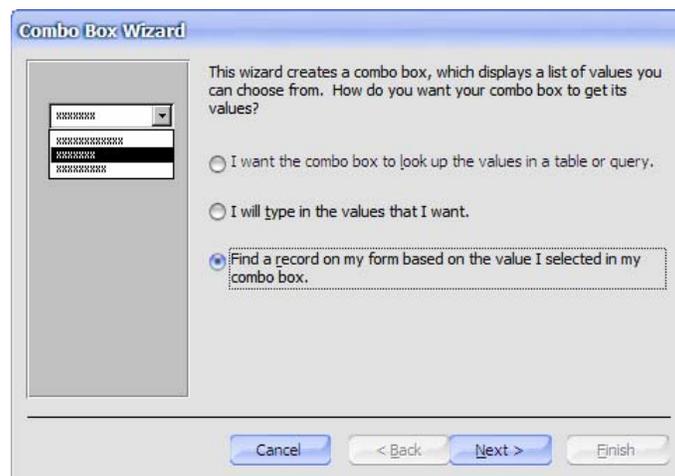
- 1) Open the *CD Entry* form in design view.
- 2) Resize the Form Header area so that it is about half a cm high.



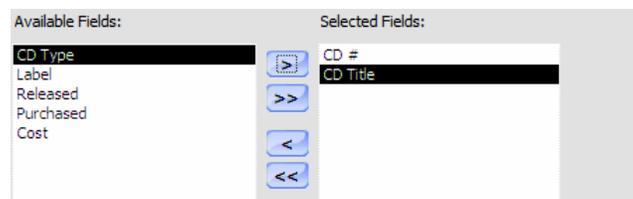
- 3) Make sure the **Control Wizards** icon in the **Toolbox** toolbar is turned on or pressed in.  Don't confuse this with the similar looking **Build** icon in the toolbars at the top. When the Control Wizards icon is turned on, a wizard will appear to guide you when certain other toolbox icons are used. We want to use a wizard to assist us with the next step.
- 4) Click the **Combo Box** icon from the **Toolbox**. 
- 5) Click in the Form Header area of the form to place a Combo Box there. A Combo Box Control Wizard will begin.



- 6) In the first step of the wizard, choose the third option; **Find a record on my form based on the value I selected in my combo box.**
- 7) Click **Next** to continue.

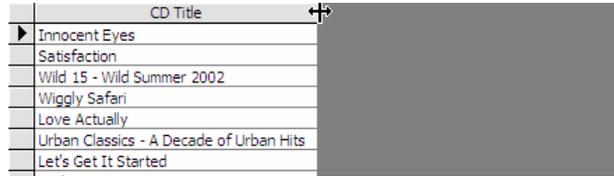


- 8) In the next step, select the *CD#* and *CD Title* fields as shown.

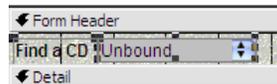


- 9) Click **Next** to continue.

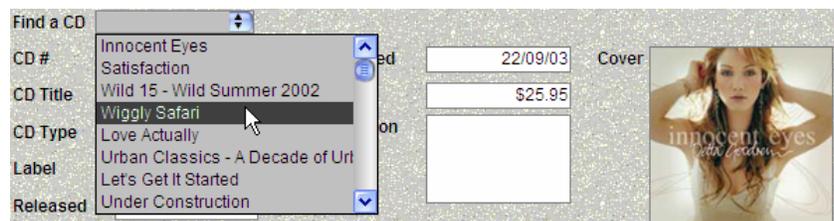
- 10) In the next step, make sure the **Hide key column** option is selected. This will mean that the list will be used to select CDs by *CD#* but only the *CD Titles* will show in the list to make it easier on the user.
- 11) Resize the list as shown. This will determined the with of the list when our combo box is complete.



- 12) Click **N**ext to continue.
- 13) In the last step, enter *Find a CD* for the label and then click **F**inish.
- 14) Resize and move the combo box and label so that they are closer together as shown.



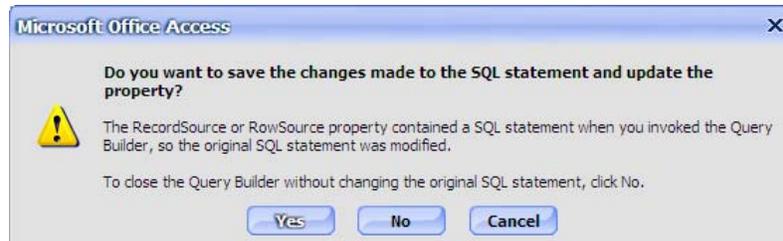
- 15) Click the **V**iew button  to preview the form.



- 16) Test the new combo box at the top of the form. When you select a CD title from the list, the details for that CD will display in the form.
- 17) Click the **V**iew  button to return to **Design View**. Lastly, we'll sort the list in alphabetical order. This will make it easier for the user because they will be able to quickly move through the list by typing the first letters of the CD title they want.
- 18) Select the combo box and make sure the properties window is displayed.
- 19) Click in the **Row Source** property. You may remember earlier it was stated that Queries are based on **SQL** (Structured Query Language). This property is currently showing an example of an SQL statement. This SQL statement can be modified in a query design window just as your queries can.
- 20) Click the **B**uild icon  to the right of the **Row Source** property. This will display a query design window, which allows you to easily edit information about what appears in the combo box's list.
- 21) When the **Query Design** window appears, change the sort order of the *CD Title* field to *Ascending* as shown on the next page.

Field:	CD #	CD Title
Table:	CDS	CDS
Sort:		Ascending
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- 22) Close the **Query Design** window.



- 23) When prompted to save changes to the SQL statement click **Yes**. You will be returned to the form and the properties window.
- 24) Preview the form and test the combo box again.
- 25) Close the form, saving changes.

Exercise 10. Adding Control Buttons to a Form

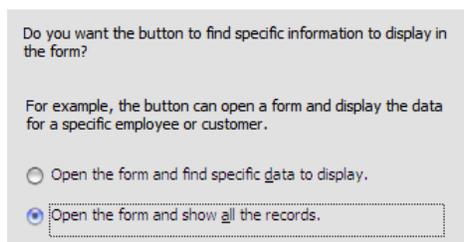
- 1) Open the *CD Entry* form in **Design View**.
- 2) Make sure the **Control Wizards** icon  is still turned on (pressed in).
- 3) Click the **Command Button** icon  in the **Toolbox** toolbar.
- 4) Click in the form header just to the right of the combo box. A Control Button Control Wizard will begin.



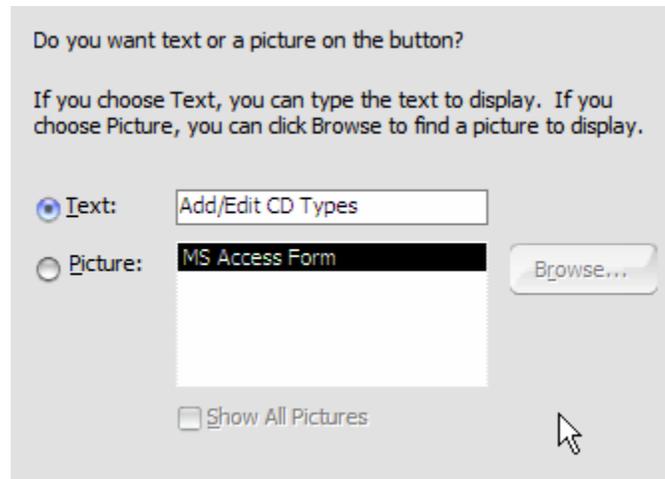
- 5) For the first step of the wizard, choose *Form Operations* from the left list and then *Open Form* from the right list as shown above.
- 6) Click **Next** to continue.



- 7) When the list of the forms in your database appears, select the *CD Types Form* and click **Next** to continue.



- 8) For the next step, make sure **Open the form and show all the records** is the option selected and click **Next** to continue.



- 9) For the next step we can choose whether to put a picture or a text caption on the form. Select the **T**ext option and then enter *Add/Edit CD Types* for the text.
- 10) Click **N**ext to continue.
- 11) For the last step in the wizard, enter *cmdEditCDTypes* for the button name and then click **F**inish.
- 12) Click the **V**iew button  to preview the form.
- 13) Click the new **Add/Edit CD Types** button. The *CD Types* form will open.
- 14) Close the *CD Types* form and then return to design view in the main form.
- 15) Follow the steps above to add a second button next to the one you just made. This one needs to open the *Genres Form*. The name property of the button should be *cmdEditGenres*.



- 16) Click on the **Control Wizards** icon  on the toolbox to turn it off since we will now add a command button without the wizard.
- 17) Place a new command button next to the other two. This time the wizard will not start.
- 18) Make sure the new button is selected and that the properties window is displayed.
- 19) Change the following properties for the new button.

Property	Change to:	Explanation
Name	cmdClose	
Caption	&Close the Form	The “&” in a caption means that the following letter will be underlined in the button. That will allow the user to activate that button by holding down the [Alt] key and pressing the underlined letter which in this case would be [Alt][C] .
On Click	Close Window	When the button is clicked, it will activate the Close Window macro we created earlier and the form will close.
ControlTip Text	Close This Form	Provides a pop-up caption when the mouse is over the button. Similar to the tips on toolbar icons – You may also like to add ControlTip Text for your other 2 buttons.

20) Resize and arrange the buttons so they appear as in the example below.



- 21) Click on the **Save** icon  to save changes to the form.
- 22) Click the **View** button  to preview the form.
- 23) Click on the Add/Edit Genres button. If it is working correctly, the *Genres Form* will open.
- 24) Close the *Genres Form* to return to the main form.
- 25) Click on the new **Close the Form** button. The form will close.

Note When you use either of the additional forms to add a new Genre or CD Type, the lists in your main form might not update with the changes until the next time you open that form.

Database Maintenance

Windows includes a number of utilities to keep your computer running smoothly. Examples of these utilities are **Scandisk** or **Check Disk** (depending on the version of Windows) - which can check disks for errors and **Defrag** - which re-arranges files more efficiently on a disk. Access also includes similar utilities. Older versions of Access included a **Compact Database** utility and a **Repair Database** utility. In newer versions of Access they are combined as the **Compact and Repair** option but they will be explained here as separate features.

The **Repair Database** utility was designed to fix errors in the database file and is comparable to the Windows Scandisk utility. Such errors could occur if the database has been closed incorrectly (such as when the database is open during a power failure or the computer crashes) and can also arise from general database use. These errors can result in the database randomly crashing or running slowly. The Repair Database utility can fix such problems reliably. It is a good idea to run it on a regular basis as this will often detect and fix errors that have not yet begun to cause problems.

The **Compact Database** utility will reorganise all of the data in a database so that it is stored more efficiently. It is similar to the Windows Defrag utility. Consider the example of packing a suitcase. If you throw things in carelessly, you won't be able to fit much in and it will take longer to find things in there. If you pack things neatly, everything can fit in less space and will take less time to find. Likewise, using the Compact database utility on your database file makes the database store data more efficiently so that the database file will be smaller and will run faster. Since general database use can clutter things up, it is a good idea to use this utility on a regular basis. It can often more than halve the size of the database file which could be the difference between fitting on a floppy disk and not fitting.

Exercise 11. Running Compact & Repair on a Database

Before commencing this exercise, you might like to open Windows Explorer (you can use the **[F][E]** shortcut) to check the current file size of your database.

- 1) From the **T**ools menu, select **D**atabase Utilities and then **C**ompact and Repair Database. If you don't already have the database open, you will be prompted for the location of the file. The database will briefly close while the utility does its work. Then the database will be re-opened.
- 2) You may want to check the file size of your database afterward to see how much difference it has made.

Note The bigger the database, the longer it will take to run the **Compact and Repair** utility.

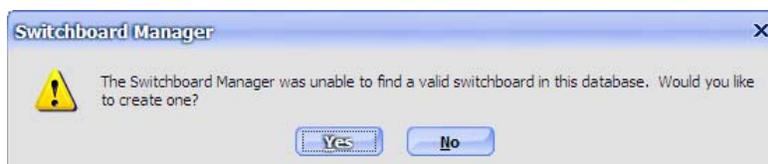
Creating an Interface

Now it's time to put the final touches on the database by creating a user friendly interface. We could create a front end by creating a blank form and manually adding command buttons to access the various objects in the database. We will do it instead by using the simple **Switchboard Manager**. We will also use the Access **Startup Options** to change what happens when the database is opened.

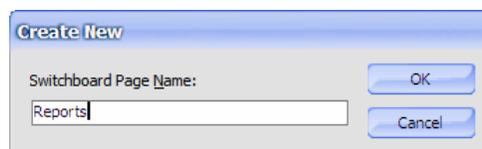
Exercise 12. Creating a Switchboard

A switch board is a database menu that consists of two parts. A form, a table containing the menu options that will appear on the form. The two components of a switchboard can be edited directly, such as formatting the form, though it is generally best to limit your changes to the **Switchboard Manager**.

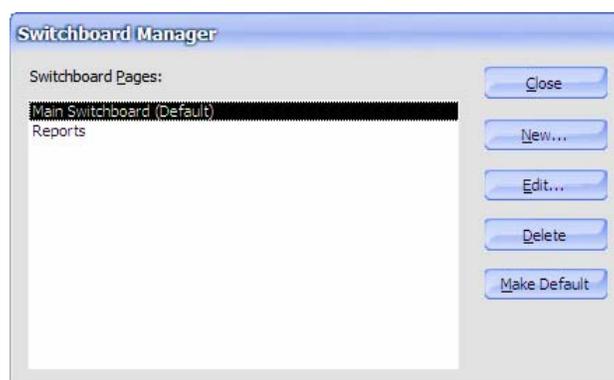
- 1) From the **T**ools menu, select **D**atabase Utilities and then **S**witchboard Manager.



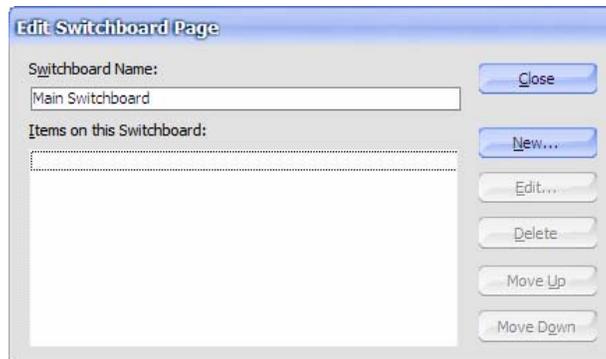
- 2) Because there is no existing switchboard, Access asks you if you want to create one. Click **Yes**. A new main switchboard will be created and we will use this as our main menu.
- 3) We also want to create another switchboard to be used as a Reports sub menu so click **N**ew to create a second switchboard.



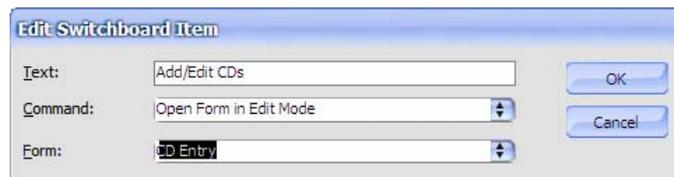
- 4) Enter *Reports* for the name of the new Switchboard. The Switchboard Manager will display both switchboards.



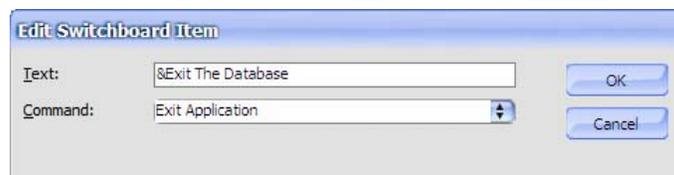
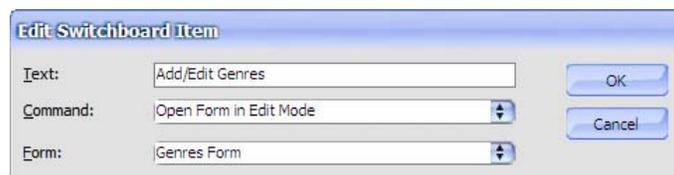
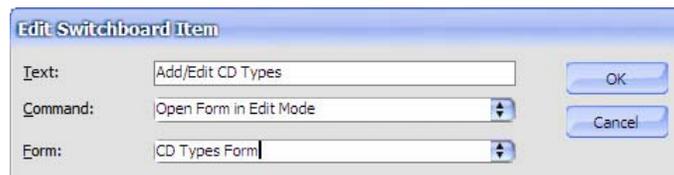
- 5) Make sure the *Main Switchboard* is selected and click **E**dit. Now we can begin to add menu items to this switchboard.



- 6) We'll start by adding a menu option which will open our main editing form. Click **New**.



- 7) Complete the **Switchboard Items** dialog as shown above and then click **OK**.
8) Add additional **Switchboard Items** as shown below.





- 9) Click **Close** to return to the **Switchboard Manager**.
- 10) Now to create the items on the *Database Reports* Sub menu. Select *Reports* from the main **Switchboard Manager** and click **Edit**.
- 11) Add the following Menu Items.



Edit: Switchboard Item

Text: Song search by year

Command: Open Report

Report: Song search by year

OK Cancel



Edit: Switchboard Item

Text: Songs listed by Genre

Command: Open Report

Report: Songs listed by Genre

OK Cancel



Edit: Switchboard Item

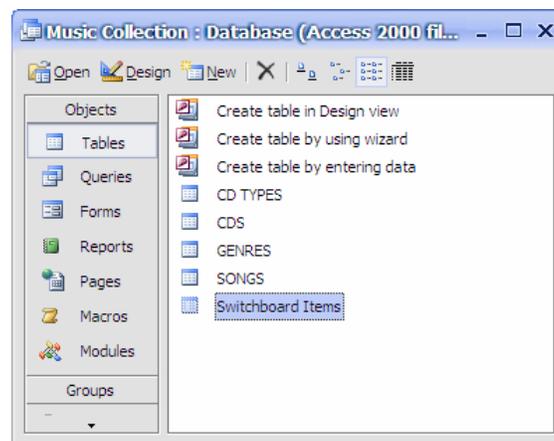
Text: Return to Main Menu

Command: Go to Switchboard

Switchboard: 1

OK Cancel

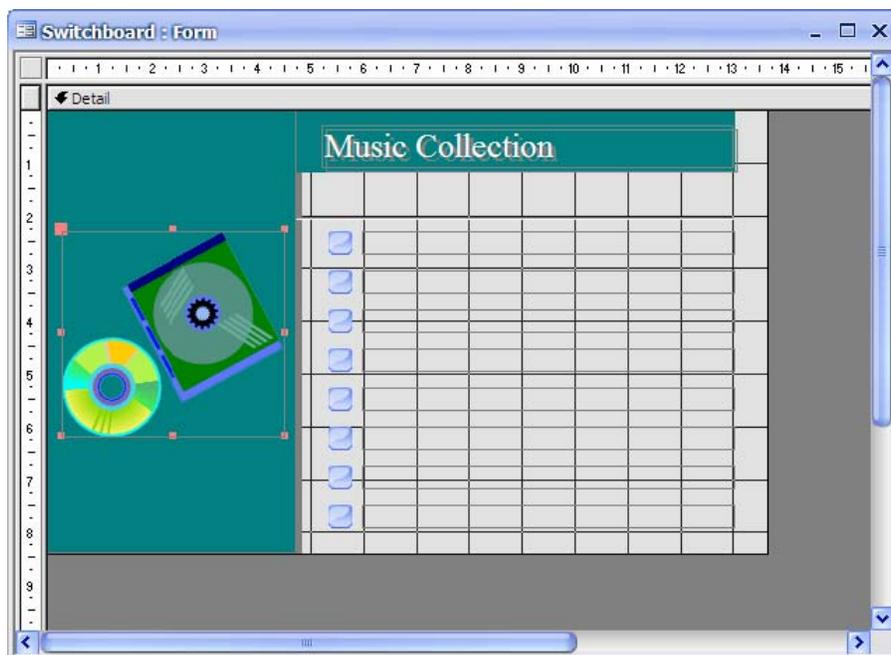
- 12) Click **Close** twice to exit the **Switchboard Manager**.



- 13) A new form has been created for your switchboard and a new table has been created to store the switchboard menu items.
- 14) Open the *Switchboard* form to test it.



- 15) Click on the **Add/Edit CDs** option. Your main editing form will open.
- 16) Click the **Close the Form** button to return to the switchboard.
- 17) Click on the **Database Reports** option to change to the sub-menu.
- 18) Click on some of the different options to preview the reports in the database and close each report after previewing it.
- 19) Click the **Return to the Main Menu** option.
- 20) Close the switchboard.
- 21) Open the *Switchboard* form in **Design View**.
- 22) Click the **Image** icon  on the **Toolbox**.
- 23) Click in the blank area to the left of the switchboard's menu to place a picture.
- 24) When the **Insert Picture** dialog appears, select the *cds.gif* file (or another suitable picture) and click **OK**.
- 25) Position the picture neatly in the middle of that area as shown.

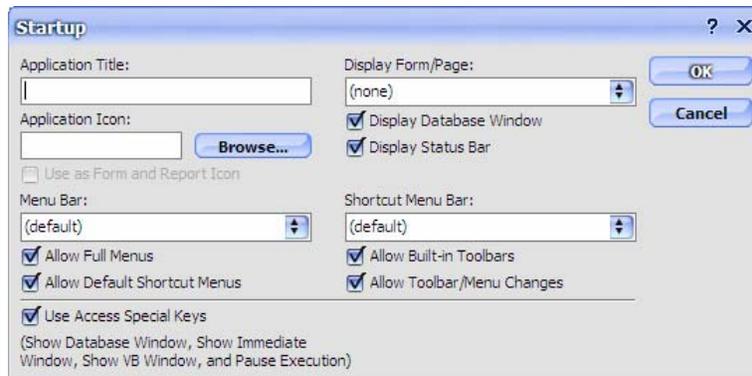


- 26) Save and close the form.

Exercise 13. Setting Database Startup Options

The **Startup** options are used to set what happens when the database opens.

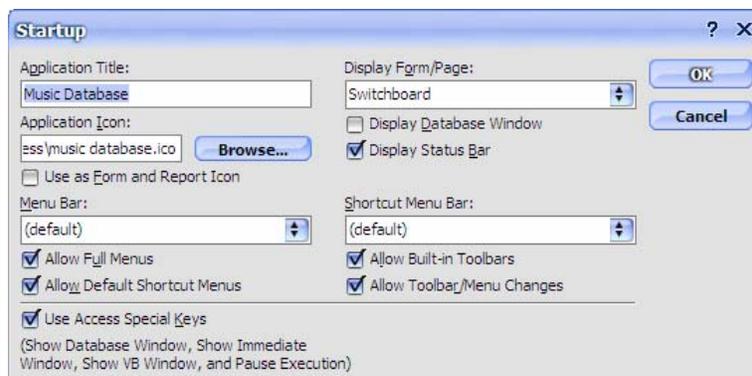
- 1) From the **Tools** menu select **Startup**.



- 2) Make the following changes in your Startup options.

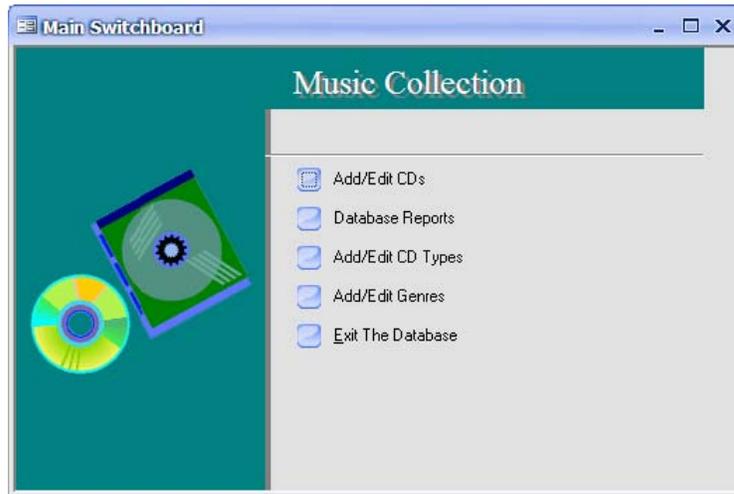
Property	Change to:	Explanation
Application Title	Music Database	Displays the title at the top of the window instead of the name of the program.
Display Form/Page:	Switchboard	The selected form will open as soon as the database opens. In this case we want our menu to be the first thing to appear.
Application Icon	music database.ico	You can browse for a custom icon to be used at the top of the screen when the database is open instead of the usual access icon.
Display Database Window	Uncheck this option	The database window will not display when the database opens. This avoids the problem of inexperienced users accidentally removing tables or other crucial objects. Remember you can always display it when you need it by pressing [F11] .

- 3) When these options have been changed, check that they appear as below.

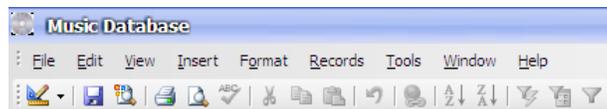


- 4) Click **OK** to confirm these settings.

- 5) Close the database and then re-open it. Your Switchboard will open and the Database Window will not appear.



Notice also that the application name and icon both appear at the top of the window.



- 6) Test some of the menu options and then click **Exit The Database**.

Tip You can also specify additional actions to take place when the database is opened by placing them in a macro called *Autoexec*. For example, you could create a macro that opens several forms at startup.