## Session IV

This week we will be discussing Forms, Macros, and Events.

### Forms:

Access forms are classified as either bound or unbound. A **bound form** is tied to a table or query. This type of form is used for editing, entering, and reviewing data in the underlying table or query. An **unbound form** is not tied to a table or query and is used to create an interface that provides the user controlled access to the application. An unbound form that is used for navigation to other forms and reports in a database and is referred to as a **switchboard**. The switchboard restricts the user to access only those controls available on the switchboard.

A **splash screen** is another type of unbound form. A splash screen form opens automatically when the database is opened, and its purpose is to give the user something to view while the application is loading.

### The User Interface:

The user interface is the mechanism by which the user communicates and interacts with the application. From the user's perspective, the user interface is the application. In most instances, you design the user interface assuming the end-user does not want to spend a lot of time learning how to use the application. Some design guidelines for meeting this goal are:

1. Design your interface to look and feel like Windows. For example, use option groups when you want the user to select only one option from a group of task related options, and use checkboxes with the user can select several options from a group of task related options. Text boxes in which the user enters or edits data should have a white background and the sunken special effects. For controls the user cannot change, use flat text boxes with the same background color as the form, or a background slightly lighter than the color of the form, but not white.

2. Simpler is better. Avoid the use of too many colors or fonts. Align controls and use rectangles to group information.

3. Be direct. Provide linear, intuitive ways to accomplish tasks. Apply tab order on forms to reinforce the visual order of the controls. Organize controls logically, based on how the user enters the data. Add keystroke shortcuts to command buttons and menu commands.

4. Provide users with feedback. Create screen tip messages. A screen tip is a message that appears when the pointer is positioned on a control on a form or on a button on the toolbar. Messages should be short and informative

5. Visual consistency is important. All forms in a database application should have the same color scheme and font properties, and the same consistent visual cues for controls. Form templates and masters help you achieve this consistency.

**Identifier operators**
Identifier operators are used in expressions, macros, and VBA code to identify objects and their properties.

The! (Exclamation point) is referred to as the **bang operator** and is used to separate one object from another or from the object collection. The collection is an object that contains a set of related objects. For example, the forms collection is a group of all currently open forms in the database, and the report collection is all the currently open reports.

Forms!frmTimeCards - Combines the name of object collections (Forms) and object names (frmTimeCards) to select a specific object.

Forms!frmTimeCards!cboEmployeeName – Identify specific fields in tables or controls on forms and reports.

The. (period) is referred to as the **dot operator** is used to separate an object from its property or method. We will discuss the bang and dot operators in future sessions. Txt.TotalHours.Visible = False. Combines object names with properties to set the value of a specific property.

**Events:**
An event is a specific action recognized by an object when it occurs on or with that object and for which you can define a response. Some events are the result of an action that the user takes; others are related to the retrieval or updating of data. Events include mouse clicks, clicking a button, opening or closing a form, keystrokes in text boxes, record updates and more. Each event has an associated event property, which can be located in the property sheet for form, reports, form and report sections, and form and report controls. Tables and queries do not have event properties. You can specify a macro name or the name of a VBA procedure in the event property of an object, and that macro or procedure runs when the event occurs. The use of event properties to run macros or to execute VBA code is called **Event-Driven Programming**.

Access automatically creates event procedures when you write event code for an object. To get to the event procedure, follow these steps:

1. Click on the object in design view and click the property sheet button on the Ribbon, or right click on the object and choose properties from the context-sensitive menu.
2. Click on the event properties tab.
3. Select the event for which you want to write code.
4. Select [event procedure] from the drop-down list.
5. Click on the ellipsis button, which places you in the Visual Basic Editor (VBE) within the event code of that object.

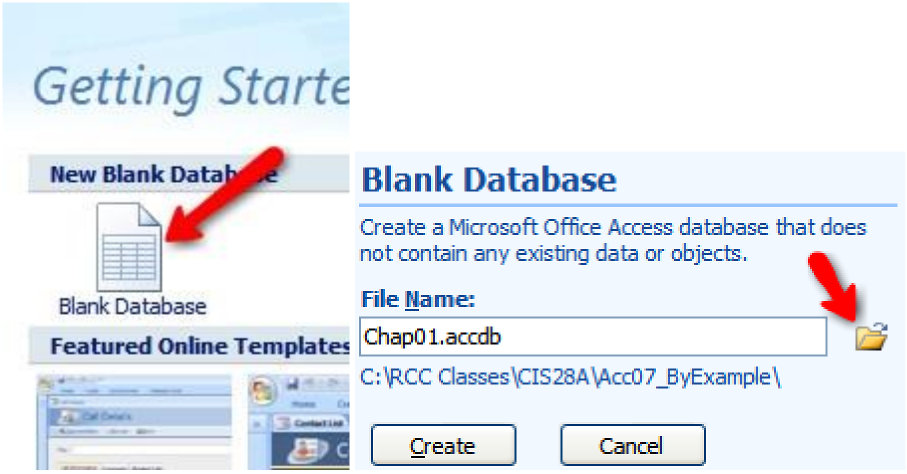
| *Event* | *Description* |
|---|---|
| **Data events** | **These events occur when data is entered, deleted, or changed in a form or control. They also occur when the focus moves from one record to another.** |
| **Error and timing events** | **These events are used for error-handling and synchronizing data on forms or reports. They include the Error event found on forms and reports and the Timer event found on forms.** |
| **Filter events** | **These events occur when you apply or create a filter on a form. The ApplyFilter and Filter events are included in this category.** |
| **Focus events** | **These events occur when a form or control loses or gains the focus. They also occur when a form or report becomes the active or inactive window. Focus is the ability to receive mouse or user input through mouse or keyboard actions. In the Windows environment, only one item at a time can have the focus.** |
| **Keyboard events** | **These events occur when you type on a keyboard. They also occur when either keystrokes that use the SendKeys macro action or keystrokes that use the SendKeys statement in VBA are sent to a form or a control on a form.** |
| **Mouse events** | **These events occur in a form or in a control on a form as a result of the mouse action, such as pressing down or clicking the mouse button.** |
| **Window events** | **These events occur when you open, resize, or close a form or report.** |

**Macros:**

Macros in Access aren’t like the macros in other Office products. You can’t record them as you can in Microsoft Word or Excel and Access does not save them as Visual Basic for Applications code. With Access macros, you can perform most of the tasks that you can manually perform from the keyboard, menu, and toolbars. Macros enable you to build logic into your application.

**Hands-on**

We will now be relying more heavily on our reference book. In order to assist you in this process you will be using the files in the CD that is in the Korol’s book.



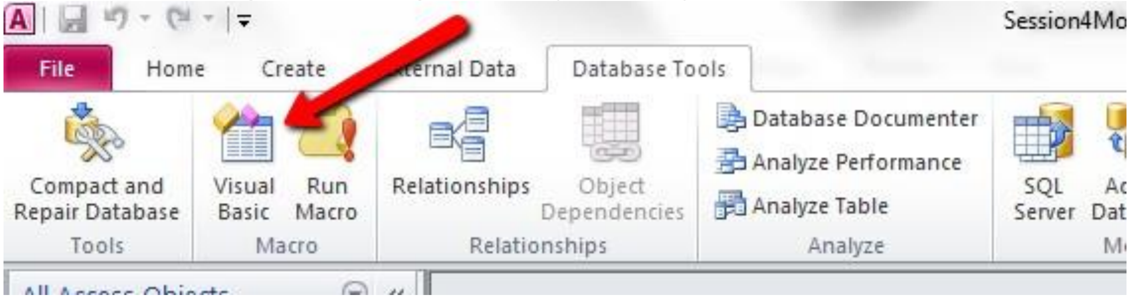
Blank Database

Create a Microsoft Office Access database that does not contain any existing data or objects.

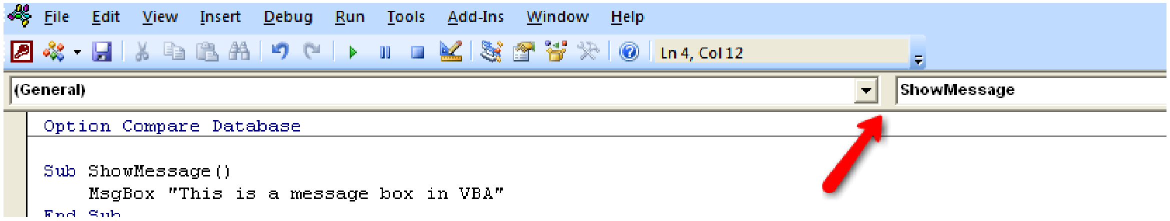
File Name: Chap01.accdb

C:\RCC Classes\CIS28A\Acc07_ByExample\

The arrow to the right is pointing to where you store your file.



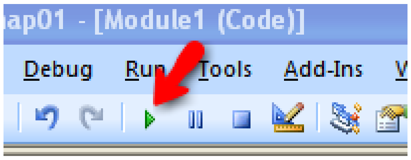
Running Show Message Routine



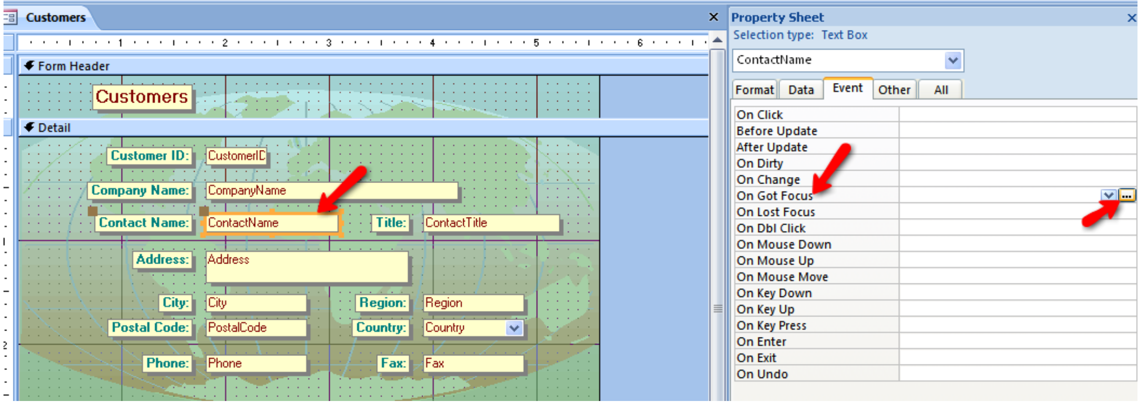
```
Option Compare Database

Sub ShowMessage()
    MsgBox "This is a message box in VBA"
End Sub
```

One of the three methods of running code



You can always close out your module by clicking the upper right X.
The Access file “HandsOn_01_3.accdb” should be under the downloaded files in blackboard or you can download the file from the site given in the book. If you can’t find the files I will send them to you.



To find the Built command you must click on the ellipses (…) the far right arrow.
If you only see one window don’t worry we will get to the other windows later.

**Assigned Readings:**

**Writing an event procedure** Korol Pg 3-7
Do Hands On 1-1 **Working in a standard module** Pg 7
Do Hands On 1-2 **Running Procedures and Functions** Pg 10
Do Hands On 1-3 **Writing an Event Procedure** Pg 15

In order to customize your database applications or to deliver products that feature user's specific needs, you'll be doing quite a bit of event driven programming. Access 2016 is

An event driven application. This means that whatever happens in an Access application is the result of an event that Access has detected. Events are things that happen to objects, and can be triggered by the user or by the system, such as clicking a mouse button, pressing a key, selecting an item from the list, or changing a list of items available in a list box. As a programmer, you will often want to modify the applications built in response to particular event. Before the application processes the user's mouse clicks and key presses in the usual way, you can tell the application how to react to the activity. For example, if the user clicks the delete button on your form, you can display a custom delete confirmation message to ensure that the user selected the intended record for deletion.

For each event defined for a form, form control, or report, there's a corresponding event property. If you open any Microsoft Access form in design view and choose properties in the tools section of the design tab, and then click the event tab of the properties sheet, go see a long list of events your form can respond to.

Forms, reports, and the controls that appear in them have various event properties you can use to trigger desired actions. For example, you can open or close a form when a user clicks a command button or you can enable or disable controls when the form loads.

To specify how a form, report, or control should respond to events, you can write event procedures. And your programming code, you may need to describe what should happen if a user clicks on a particular command button or makes a selection from a combo box. For example, when you design a custom form you should anticipate and program events that can occur at runtime (while the form is being used). The most common event is the click event. Every time a command button is clicked, an event procedure to responds the click event for that button.

When you sign your event procedures to an event property, you set an event trap. If entrapping gives you considerable control and handling events because you basically interrupted default processing that access would normally carry out in response to the user's key press or mouse click. If a user clicks a command button to save a form, whatever code you've written in the click event of that command button will run. The event programming code is stored as a part of a form, report, or control and is triggered only when user interaction with a form or report generates a specific event; therefore, it cannot be used as a stand-alone procedure.

**Why use events?**

By using events you can make your applications dynamic and interactive. To handle a specific event, you need to select the appropriate event property on the property sheet, and write an event handler procedure. Access will provide his own default response to those events you have not programmed. Events cannot be defined for tables and queries.

**Placing an Access database in a trusted location** Korol Pg 22

The new security features in access 2016 make it more complicated for users to work with databases that include VBA code and other executable content such as Action queries that add, modify, or delete data, DDL (data definition language) queries that create and alter tables, SQL pass-through queries, and action X controls, as well as certain unsafe expressions and macro commands.

When an Access database is open Access 2016, by default, Access turns off all potential unsafe code or content and displays a security warning messages just under the ribbon.

To use the disabled components of the database the user must click the Options button on the message bar. This will bring up the security alert dialog box. By default, the “Help protect me from unknown content (recommended) option button is selected. To enable the disabled content only for the current session, users need to select the "enable this content” option button and then click ok. These steps will need to be repeated each time the database is open. If users select not to trust the database -- in other words, they ignore the security warning message -- they will still be able to view the data in the database and use components that Access has not disabled.

To make it easy to work with access databases in this book, you will not want to bother with enabling content each time you open a database. To trust your database is permanently, you can put them in a trusted location -- a folder on your local or network drive that you mark as trusted.

Notice the Open the Trust Center hyperlink at the bottom of the Security Alert dialog box. This hyperlink will open the trust center where you can set up a trusted folder.

**Using the Auto Exec Macro** Korol pg. 869 - 871
Do Hands-On 29.1 **Understanding and Using the Auto Exec Macro** Pg. 870
The most important macro that every Access programmer needs to be familiar with is the Auto Exec macro. This macro is not new to Access 2016; it's been with Access since the very beginning. An Auto Exec macro in your Access application will automatically run when the database is open. This is very convenient, especially when you need to check whether the rest of your application will load.
**Creating and Using Macros in Access 2016** Korol Pg. 874 - 881

---

**Project**

Before leaving the company Jason had created some of the data entry forms for the MovieCam Technologies database. At this point, it would be best to develop a **form template** you can use to design the rest of the forms. You can apply all the common properties that are characteristic of the forms to the form template, and then build individual forms using the template’s basic structure and design. You can also create a **form master**, which can contain items such as a company logo, company name, and other standard controls that you want to be part of any newly created form or switchboard.

The design of the switchboards will be different from the design of other forms in the database, so you'll need a different form master as well. The switchboard will contain the logo and company name.

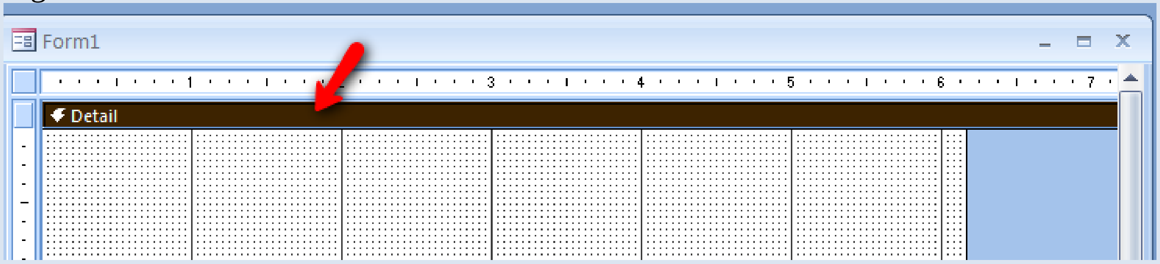
### To Create a Form Template

**Step 1**
Start Access and load the Session4Movie database.

**Step 2**
In the **Ribbon** under the **Create** tab select **Form Design**.

**Step 3**
Maximize the form to make it easier to work with.

**Step 4**
Right click on the **Detail** section of the form. Select **Form Header/Footer**.

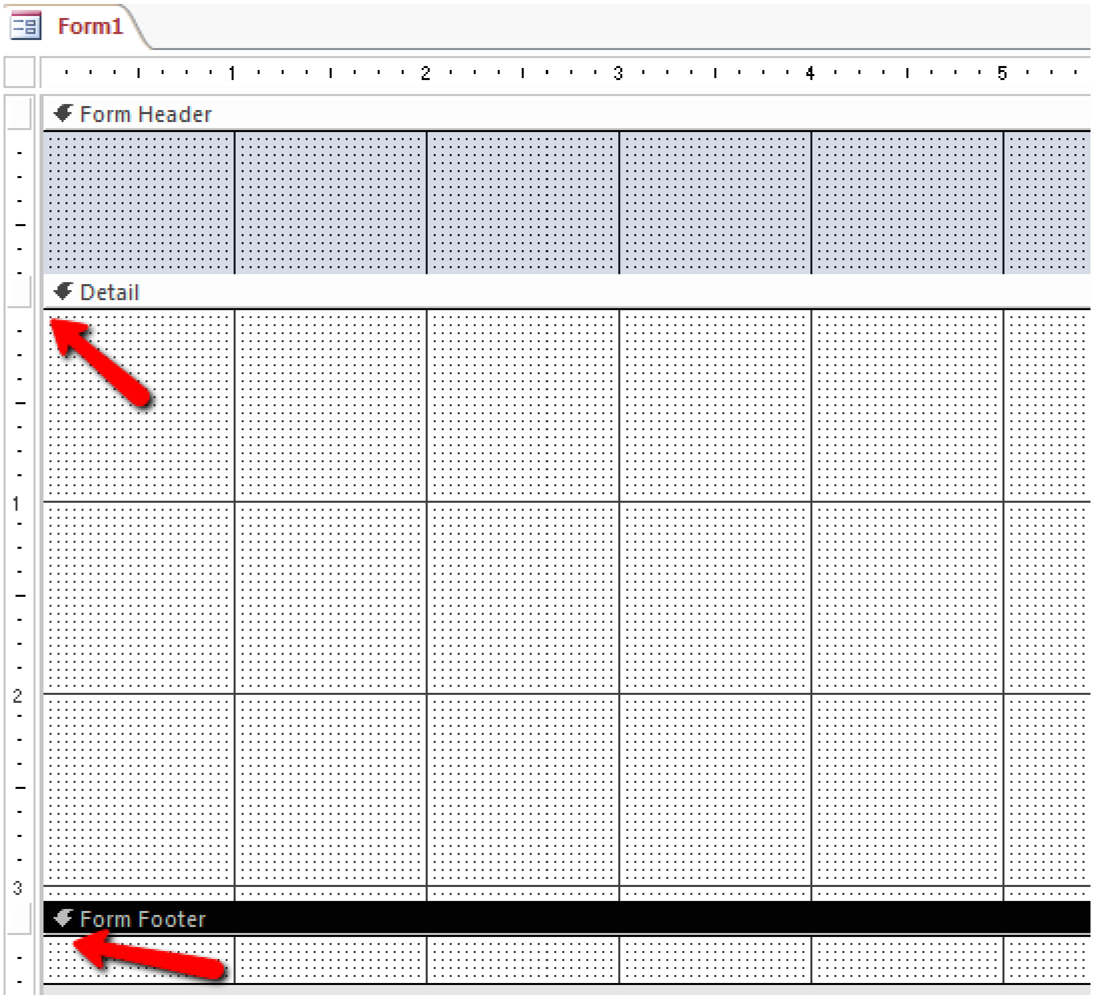


**Step 5**
Position the pointer at the bottom edge of the **Form Header** section (your cursor is on the Detail section) until it changes to a double arrow shape, and then drag the Detail section down to the ¾ "mark on the vertical ruler.

**Step 6**
Position the pointer on the right edge of the form until it changes to a double arrow shape, and then drag to the 6 ½ "mark on the horizontal ruler.

**Step 7**
Position the pointer at the top edge of the Form Footer section until it changes to a double arrow shape, and then drag to the 3"mark on the vertical ruler.

**Step 8**
Position the pointer at the bottom edge of the Form Footer section until it changes to a double arrow shape, and then drag to the ½ "mark on the vertical ruler.



**Step 9**
Scroll to the top of the form, and then select the form by clicking the **form selector**, which is the box in the upper-left corner of the form where the two ruler bars meet. Note that when a form is selected a small black square appears in the middle of the form selector. The form selector is not a toggle button; clicking it again does not deselect the form.

**Step 10**
Click the properties button on the Form Design toolbar, and then click the **All** tab on the Property sheet if necessary.

**Step 11**
Scroll down the list of properties, and then change the Record Selectors property to **No** and if necessary the Dividing Lines property to **No**. (You can double click on the property to change the value) Record Selectors are not necessary in Switchboards and Dividing Lines show the end of one form section and the beginning of another.

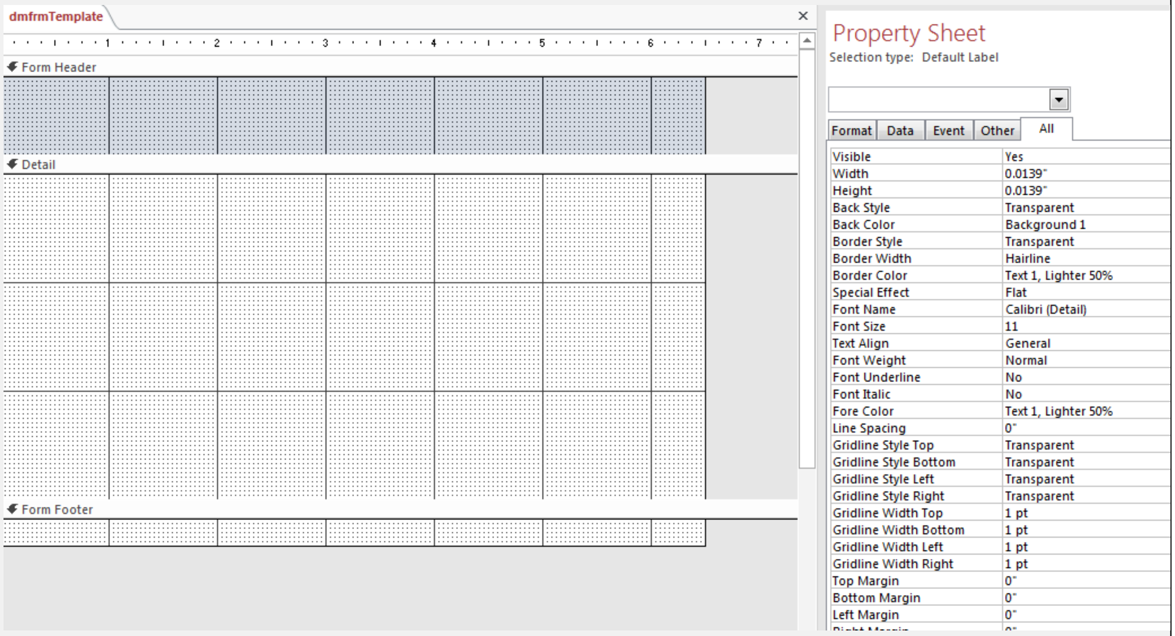
**Step 12**
Close the form and save it as dmfrmTemplate. The tag dm is used to identify the form as part of the development and maintenance of your database application. Users won’t be working with this form so we will hide it at the end of our development.

**Step 13**
Find the dmfrmTemplate in the Navigation Box, rt_click the form to open in Design View

## Changing the Default Properties of controls:

**Step 1**
With form dmfrmTemplate still open.

**Step 2**
On the Ribbon select the **Design Tab** and Click the **Label** button in the **Control** group(Aa) The default label properties are now listed in the property sheet. **See below**



Property Sheet
Selection type: Default Label

Format | Data | Event | Other | All

| | |
|---|---|
| Visible | Yes |
| Width | 0.0139" |
| Height | 0.0139" |
| Back Style | Transparent |
| Back Color | Background 1 |
| Border Style | Transparent |
| Border Width | Hairline |
| Border Color | Text 1, Lighter 50% |
| Special Effect | Flat |
| Font Name | Calibri (Detail) |
| Font Size | 11 |
| Text Align | General |
| Font Weight | Normal |
| Font Underline | No |
| Font Italic | No |
| Fore Color | Text 1, Lighter 50% |
| Line Spacing | 0" |
| Gridline Style Top | Transparent |
| Gridline Style Bottom | Transparent |
| Gridline Style Left | Transparent |
| Gridline Style Right | Transparent |
| Gridline Width Top | 1 pt |
| Gridline Width Bottom | 1 pt |
| Gridline Width Left | 1 pt |
| Gridline Width Right | 1 pt |
| Top Margin | 0" |
| Bottom Margin | 0" |
| Left Margin | 0" |
| Right Margin | 0" |

**Step 3**
Make sure the Visible property is set to **Yes**, scroll down the list, and then change the Font Size property to **10** and the Font Weight property too **Semi-Bold**.

Now you will change the default settings for the Label control

**Step 4**
Click the **Text Box** button in the Control group in the Ribbon to display the Text box property window.

**Step 5**
Change the Font size to 10.

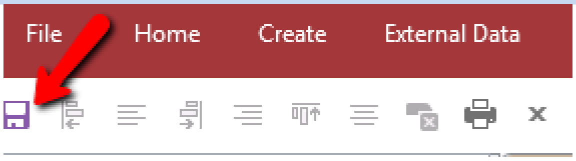
**Step 6**
Change the Label X property to **0** and the Label Y property to **-0.25**. The next time you add a Text Box control to a form, its accompanying label control will be positioned above the text box, instead of to its left. This is not always desirable, but it serves to illustrate this capability.

**Step 7**
Change the Text Align properties to **Left**. These setting will left-align the text within the text box, and left-align the text within the associated label.

**Step 8**
Change the Add Colon property to NO to eliminate the colon in the label.

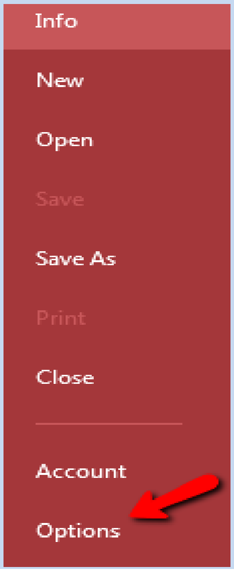
**Step 9**
Close the property sheet and click **restore down** the form window.

**Step 10**
Click the save button on the **form Designer toolbar** to save the changes you have made, and then close the form.

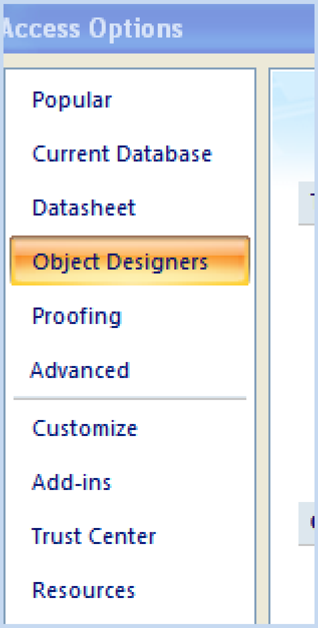


## To set the dmfrmTemplate as the default

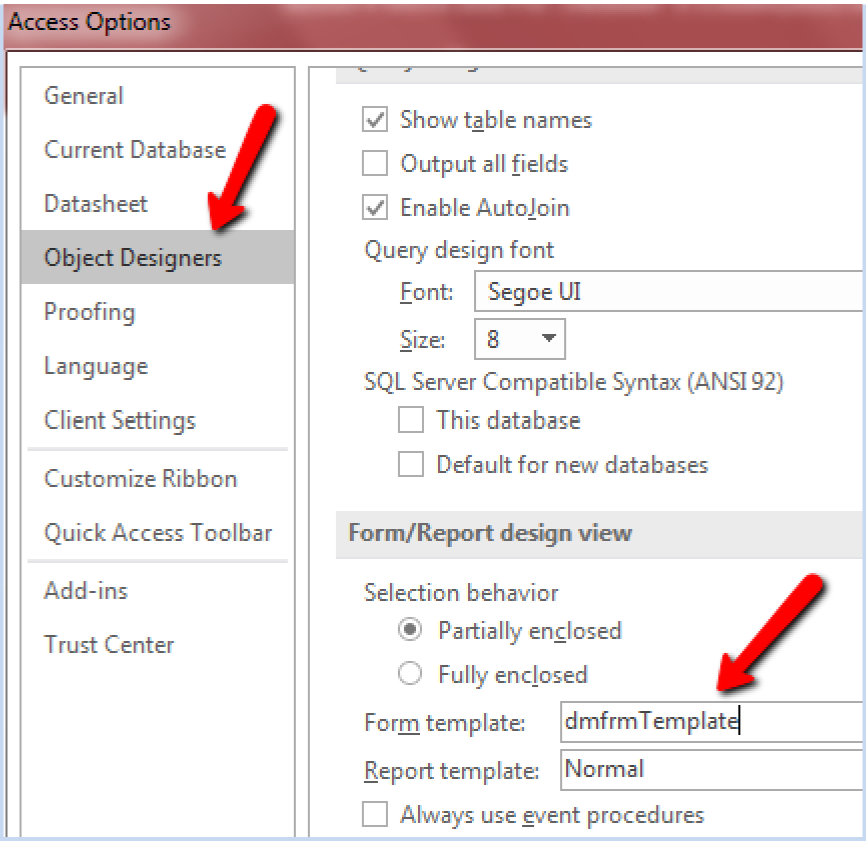
**Step 1**
Click tools on the menu bar and then click Options. The Options dialog box opens



**Step 2**
Select **Object Designers** in the Access Options Window.



**Step 3**
Type **dmfrmTemplate** in the form template and then click the OK button,



☑ Show table names

□ Output all fields

☑ Enable AutoJoin

Query design font

Font: Segoe UI

Size: 8

SQL Server Compatible Syntax (ANSI 92)

□ This database

□ Default for new databases

**Form/Report design view**

Selection behavior

● Partially enclosed

○ Fully enclosed

Form template: dmfrmTemplate

Report template: Normal

□ Always use event procedures

**Step 4**
The Options box closes, and the dmfrmtemplate form has now been set as the default.

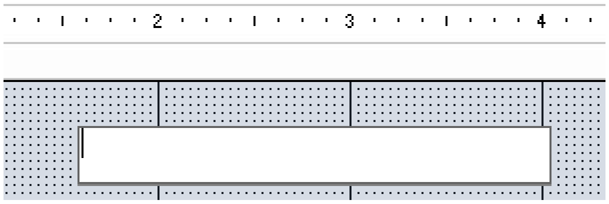
### Creating Form Masters

A custom form template represents a basic structure for all forms in the database and determines the dimensions, properties, and form sections of new forms. It can not contain any controls, however. To add the same controls to forms in a database, such as a label control for the company name or an image control for company logo, you can create a form master. A form master is a form that contains the controls that are common to all forms in the database. Unlike a template, new forms created in the database do not inherit the characteristics of the form master. However, the form master can be copied and pasted to create new forms. You decide that all data entry form should contain the Company logo and the company name.

### To create a data entry form master:

**Step 1**
Create a new form in design view. Remember use the Create tab and the Form Design icon. The form is automatically based on dmfrmTemplate.

**Step 2**
In the Design Tab, click the **label** button. Place the pointer at 1 ½" mark in the header section, hold the left button down and create a rectangle, ½ x 2 ½"

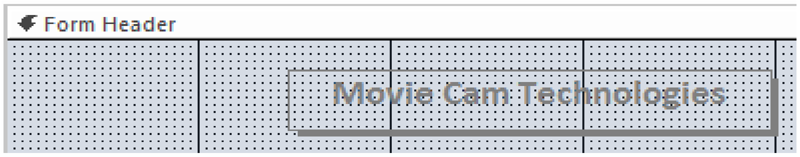


**Step 3**
Type the text **Movie Cam Technologies** inside the label control, and then press the enter key.  .

**Step 4**
Click the Font Size list arrow on the Format section in the Ribbon, and then click 14. The text should be bold, based on the Font Weight property you set for the default label control

**Step 5**
Click the Design Tab in the Ribbon and select the Property Sheet and then select Special Effect: Shadowed.

**Step 6**
Save the form as dmfrmMaster.



**Adding A Picture To The Form:**

**Step 1**
Open form dmfrmMaster in Design view.

**Step 2**
Click **Logo** in the **Design** tab under the **Header/Footer** section.
Select the logo file (logo.wmf) you downloaded from the Blackboard files for Hands On exercises into your CIS28A folder.

**Step 3**
Click in the Form Header to display the MovieCam logo.

**Step 4**
Resize and reposition the logo as shown below: Note that the size and position of the logo do not have to be exact, but as close as possible to that shown:

**Step 5**
Select both the **Image** control containing the logo and the MovieCam Technologies label.

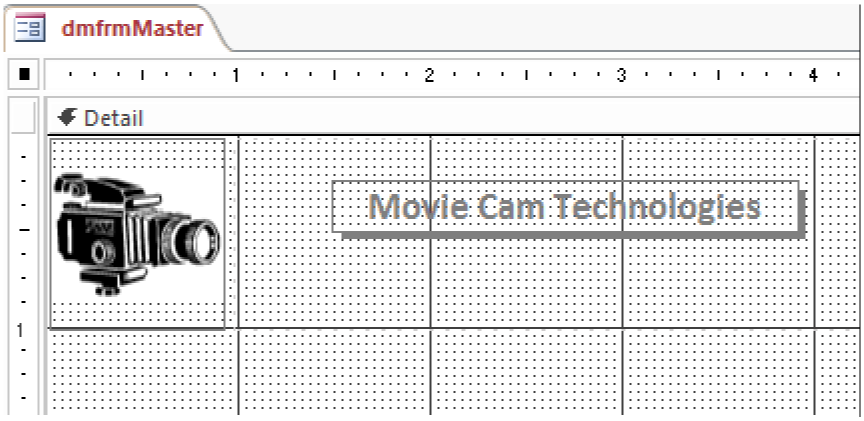
**Step 6**
Click the **Cut** button from the Home tab and click anywhere in the Detail section of the form.

**Step 7**
Click the **Paste** button from the Ribbon to paste the two controls into the Detail section.

**Step 8**
Right click on the **Detail** section and select **Form Header/Footer** from the drop down list.

This will remove the Header/Footer sections from the form. You will not need header and footer sections in the switchboard forms.

**Step 9**
Close and Save the form.



**Changing Form Properties of the dmfrmMasterSwitchboard:**

**Step 1**
Right-click the dmfrmMaster form in the Navigation Pane, and then click Copy on the short-cut menu.

**Step 2**
Right-click in an empty area in the Navigation Pane, and then click Paste and Rename the form to dmfrmMasterSwitchboard.

**Step 3**
Open dmfrmMasterSwitchboard in Design View.

**Step 4**
If necessary, select the form. The form is selected if the form selector in the upper left hand corner of the form displays a small black box.

**Step 5**
Click on the **Design Tab** and select the Property Sheet in the Ribbon
Click on the **All** tab.

**Step 6**
Change the Scroll Bars property to **Neither** because the switchboards will be designed so that all controls fit on the screen and scrolling will not be necessary.

**Step 7**
Change the Navigation buttons to **No**. Switchboards have no underlying records through which to navigate.

**Step 8**
Change the Auto Center property to **Yes** so that the open switchboard will be centered on the screen.

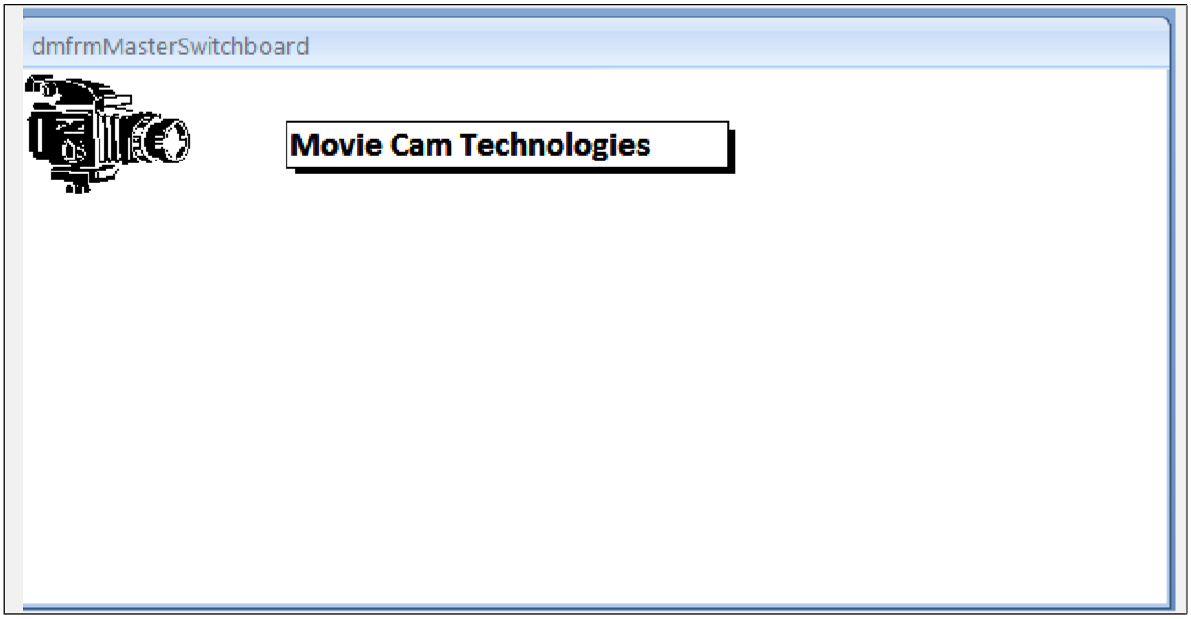
**Step 9**
Change the Border Style property to **Dialog** so that users cannot change the size of the form.

**Step 10**
Change the Control box property to **No** to remove the control box from the upper-left corner of the window.

**Step 11**
Change the Min Max Buttons property to **None** so that the user cannot minimize or maximize the form.

**Step 12**
Change the Close Button property to **No**. You want to design the switchboards so that users can navigate only where you want them to go. Eliminating the buttons for closing or minimizing the switchboards gives you more control of the database areas to which users have access.

**Step 13**
Close the property sheet, and then switch back to form view. Your form should look like the one below.



## To Create the Data Entry Forms Switchboard

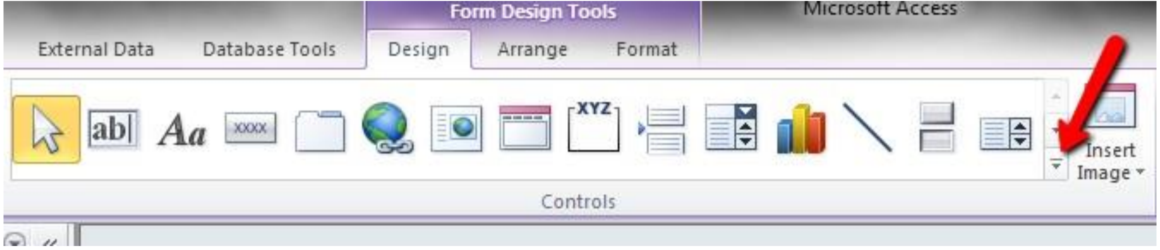
**Step 1**
Right click the dmfrmMasterSwitchboard form in the Navigation Pane, and then click **Copy** on the shortcut menu. Go to an empty area of the Navigation Pane and right click and select **Paste**.

**Step 2**
A copy of dmfrmMasterSwitchboard is created. Right-click on the form and select **Rename** and label the form frmDataSwitchboard.

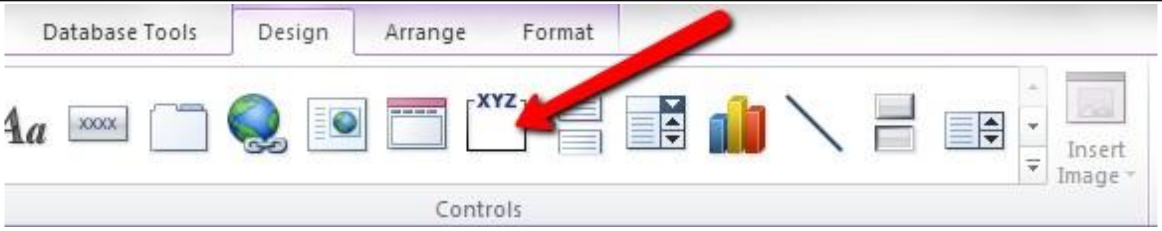
**Step 3**
Open the frmDataSwitchboard in Design view.

## Adding an Option Group to the Switchboard

**Step 1**
Make certain Use Control Wizards is selected. This is located in the control section of the design ribbon

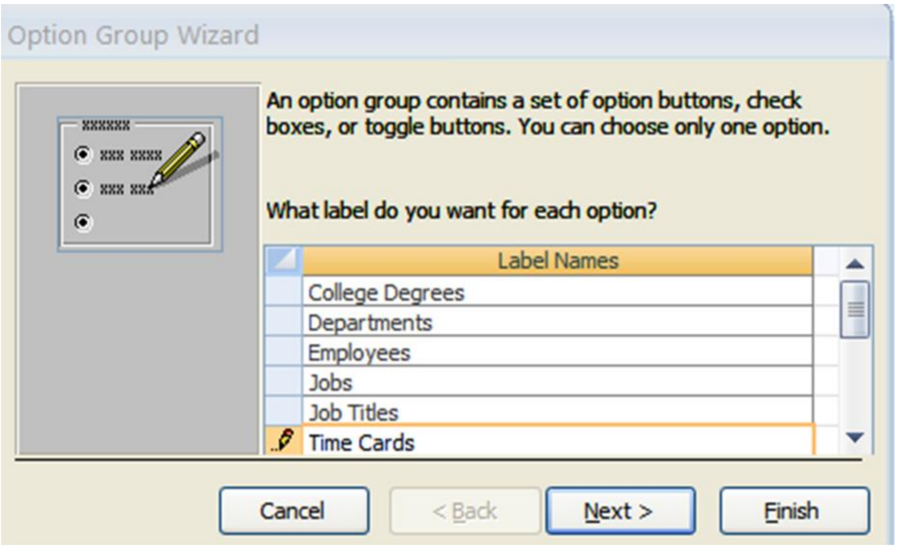


**Step 2**
Click the **Design** tab in Ribbon and select the **Option Group** button.



**Step 3**
Click anywhere in the Detail section, drag to draw an option control that is approximately 2“tall and 2“wide. The Option Group Wizard dialog box opens, and you are instructed to enter the labels, or names for each option in the option group.

**Step 4**
Type the label names as shown below, pressing the Tab key or the down arrow key to move to each text box. Each label represents the name of a data entry form in the database:



An option group contains a set of option buttons, check boxes, or toggle buttons. You can choose only one option.

What label do you want for each option?

| Label Names |
|---|
| College Degrees |
| Departments |
| Employees |
| Jobs |
| Job Titles |
| Time Cards |

**Step 5**
After you have entered all the label names, click the **Next** button. The next dialog box asks if you want the option group to show one choice as the default Click **No, I don’t want a default** option button, and then click the **Next** button.

**Step 6**
The next dialog box shows the numbers the wizard will assign to each choice in the option group. When the user clicks a button in the group, the value of the option group itself becomes the number that corresponds to the option button selected by the user. This allows you to test the value of the option group in a macro and then open the appropriate form. Click the **Next** button to accept the numbers the wizard assigned to each option and continue.

**Step 7**
The next dialog box lists several style options. You’ll accept the default etched option buttons selection, which is the standard windows style for option groups.

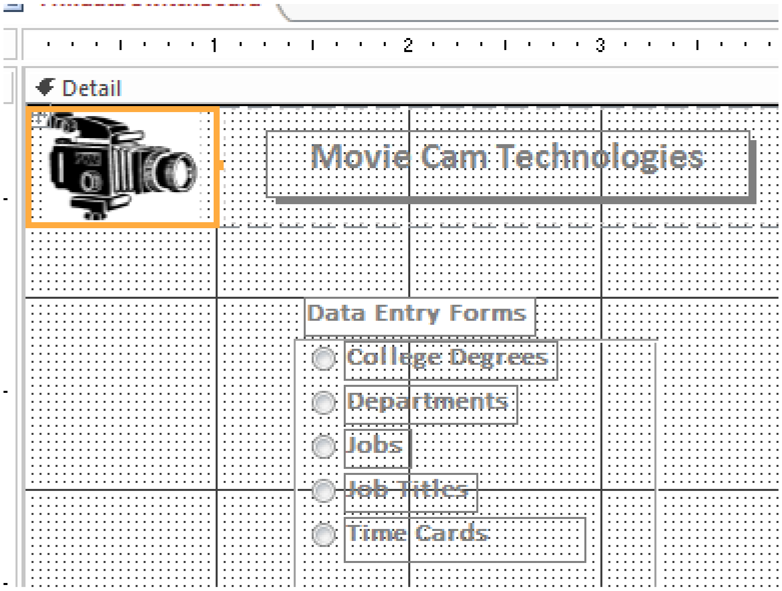
**Step 8**
Click the **Next** button to continue. In this last dialog box, you enter the caption
Data Entry Form.

**Step 9**
Click the **Finish** button. The form control with the labels appears on the switchboard.

**To Move Controls on the Form**

**Step 1**
Drag the bottom border of the detail section down so that it is approximately 4“in height.

**Step 2**
Position the pointer on the lower center handle of the option group frame, as shown below, and then click and drag down approximately ¼” to enlarge the frame slightly to better accommodate the options



**Step 3**
Drag around the option group, including its label and contents, to select them all, and then drag the selection down, so
 That the top border of the label data entry form is approximately even with the 1” mark On the vertical ruler bar and directly under, and left aligned with the Movie Cam Technologies Label

**Step 4**
Click a blank area of the form to deselect the option group, and then drag around all the options within the option group, except college degrees, to select them. Be sure not to select the option group frame or the data entry form’s label at the top.

Next you want to space the controls evenly one above another.

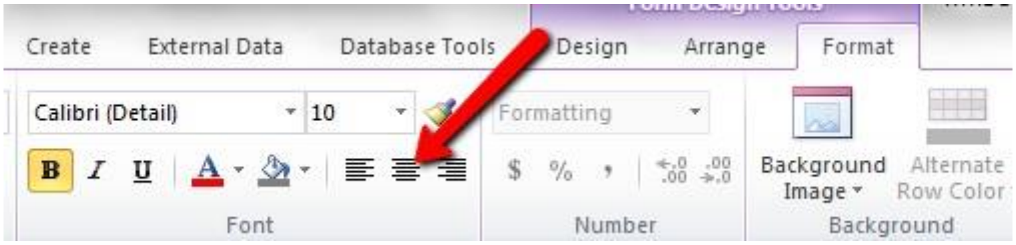
**Step 5**
Press and hold the **Ctrl** key and press the down key five times. This moves down the selected options in slight increments.

**Step 6**
Deselect the options, and then select all the options in the option group, except College degrees and Departments.

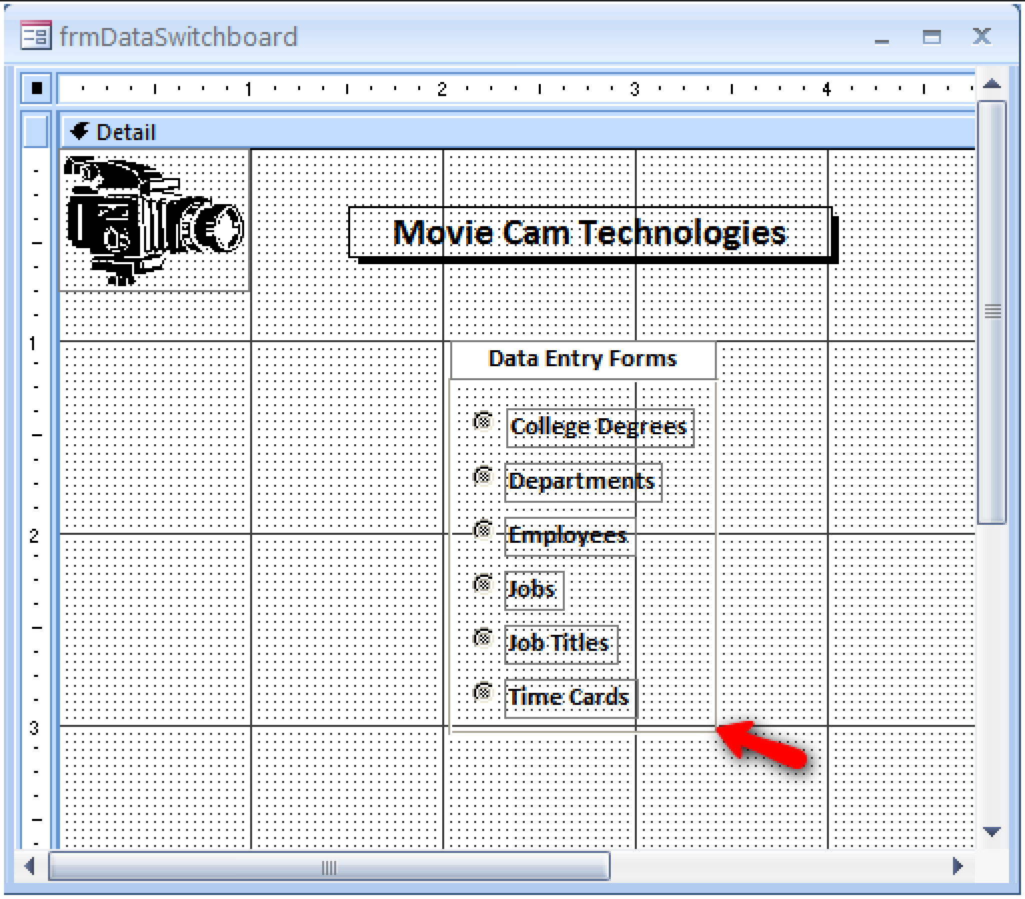
**Step 7**
Press and hold the control key and press the down key five times, and then deselect the options.

**Step 9**
Repeat this process to insert the same amount of space between the remaining options in the option group. Adjust the bottom of the frame, as needed, to leave some space between the last options (timecards) in the bottom of the frame.

**Step 10**
Click the **Data Entry Forms** label to select it, position the pointer on the upper left corner handle until the pointer changes to a finger-pointing shape, and then drag to move the label above the option group. Size a label to the width of the option group frame; click the **Center** button in the **Format** section of the **Design** tab in the ribbon to center the text within the label.



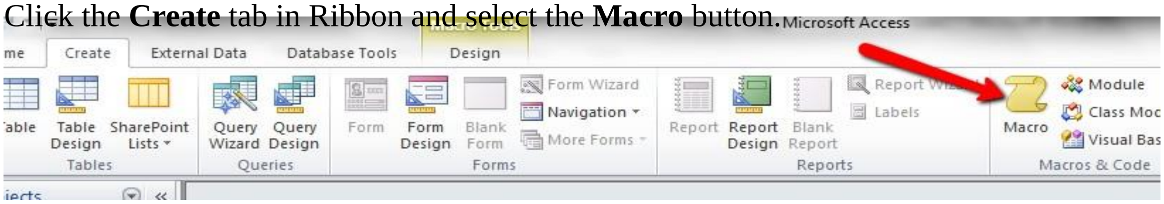
**Step 11**
Select the option group frame, click the **Properties** button in the Ribbon, Change the name property to **grpForms**, and then close the property sheet.



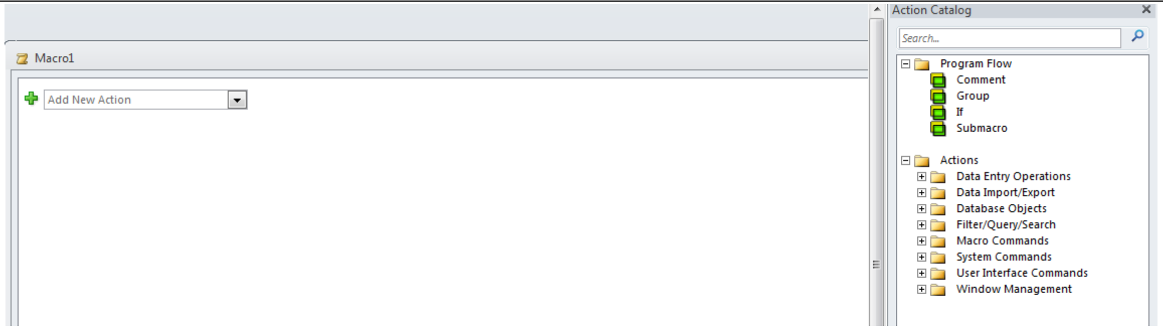
**Step 12**
Save your changes and close the form.

**Exploring a Macro Window:**

**Step 1**
Click the **Create** tab in Ribbon and select the **Macro** button.



The macro window opens as shown below:

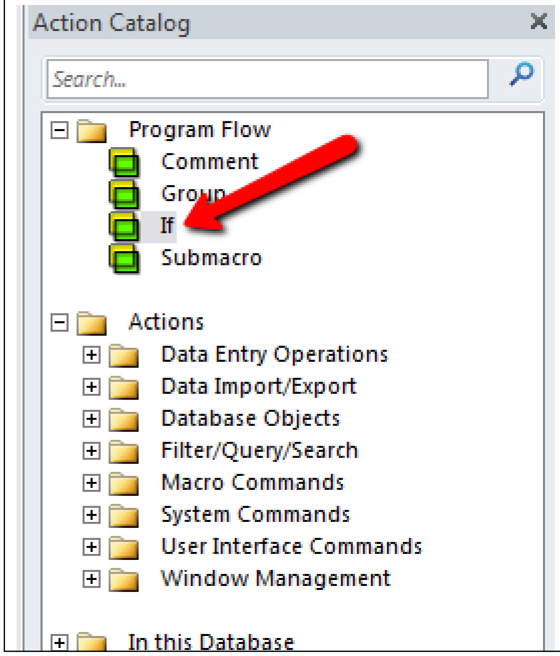


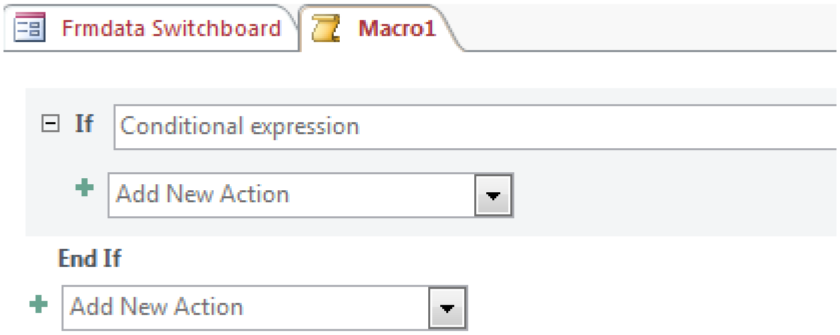
In this default setting the macro window consists of the default **MacroName** which is **Macro1** and the **Add New Action** window.

**Creating the Macro for the Option Group:**

**Step 1**
Make sure that the **Action Catalog** is open, If not click on the **Design tab** and then the **Action Catalog.**

**Step 2**
**With the action catalog visible on the right side of the screen Dbl-Click** on the **If** icon this sets up an **If Then** structure for the new macro.



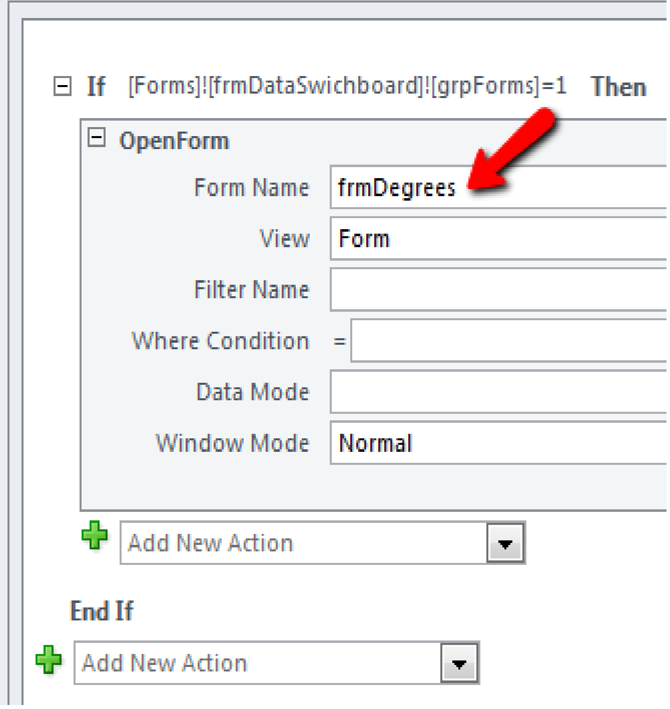


**Step 3**
Type **Forms![FrmdataSwitchboard]![grpForms]=1** next to the **If** word. This expression tests to see if the value of the group forms option group is equal to one. If it is, you want to open the frmDegrees form; recall that College degrees is the first form listed in the option group. (note if you have problems you may need to go back and recheck your work on the option group)

**Step 4**
**Press the tab key twice** to enter the **Add New Action box**. Click the action list arrow and locate **OpenForm**. Click on Openform from the dropdown list.

**Step 5**

Tab to the **Form Name** and select from the list box the form **frmDegrees**.
You will accept all the other default values.



If [Forms]![frmDataSwichboard]![grpForms]=1 Then

OpenForm

Form Name: frmDegrees

View: Form

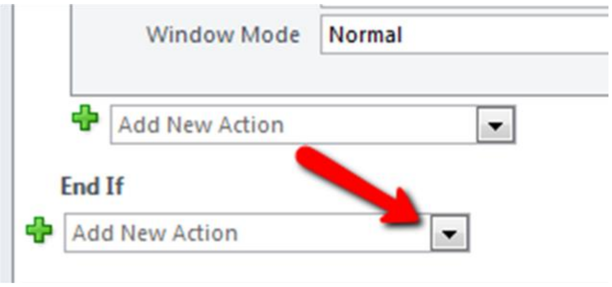
Filter Name:

Where Condition: =

Data Mode:

Window Mode: Normal

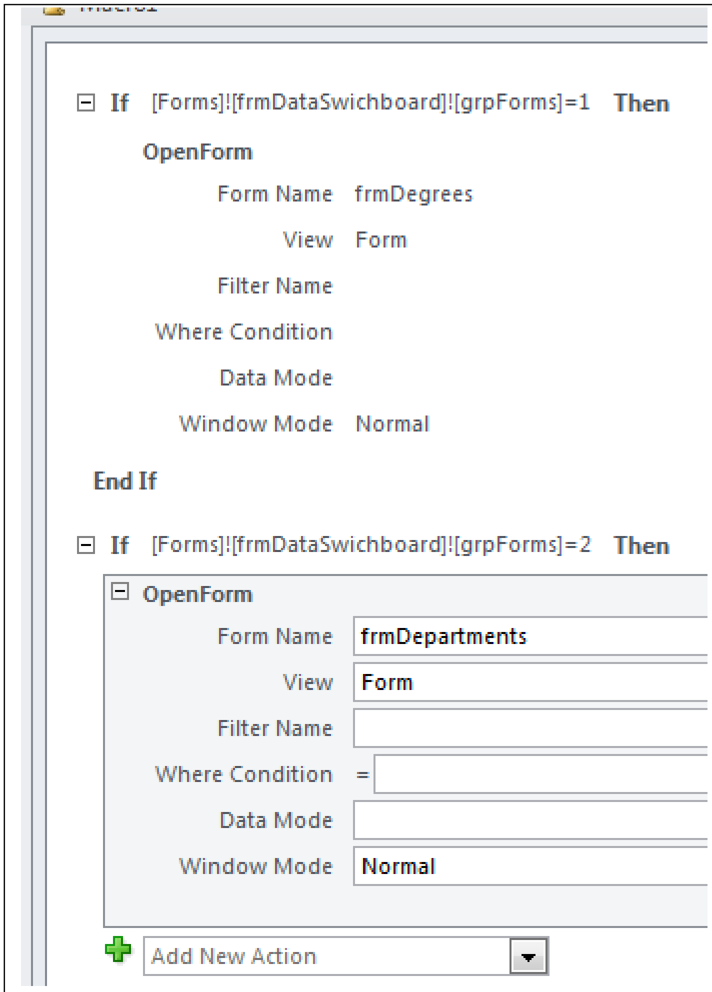
Now you need to specify the rest of the conditions. Repeat the above process but this time just select the action **If** in the **Add New Action box** which is shown by the arrow below. **<u>Be certain you begin below the End If statement</u>**!



Window Mode: Normal

**Step 6**
Starting after End If add the following 5 actions

Type **Forms![FrmdataSwitchboard]![grpForms] =2** and openform **frmDepartments**
Type **Forms![FrmdataSwitchboard]![grpForms] =3** and openform **frmEmployees**
Type **Forms![FrmdataSwitchboard]![grpForms] =4** and openform **frmJobs**
Type **Forms![FrmdataSwitchboard]![grpForms] =5** and openform **frmJobTitles**
Type **Forms![FrmdataSwitchboard]![grpForms] =6** and openform **frmTimeCards**



```
⊟ If  [Forms]![frmDataSwichboard]![grpForms]=1  Then
        OpenForm
                Form Name  frmDegrees
                     View  Form
              Filter Name
          Where Condition
                Data Mode
              Window Mode  Normal
  End If
⊟ If  [Forms]![frmDataSwichboard]![grpForms]=2  Then
      ⊟ OpenForm
                Form Name  frmDepartments
                     View  Form
              Filter Name
          Where Condition  =
                Data Mode
              Window Mode  Normal
```

**Step 7**
Close and Save the macro as **mfrmDataSwitchboard**.

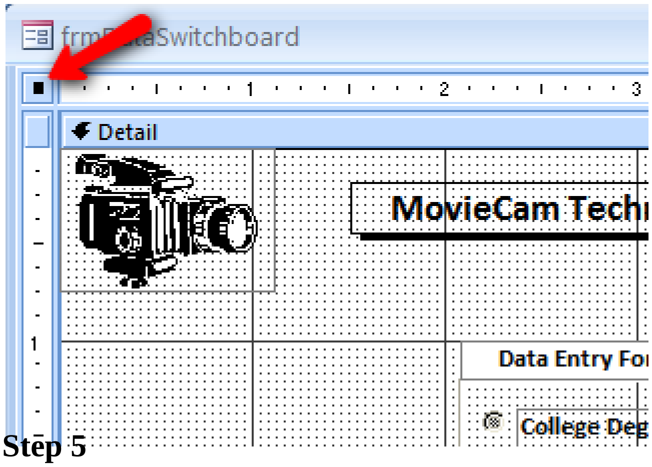
**Setting the after update property of the group forms option group:**

**Step 1**
Open the **frmDataSwitchboard** form in Design view.

**Step 2**
Click the **grpForms** option group frame, and then click the **Properties** button in the Ribbon

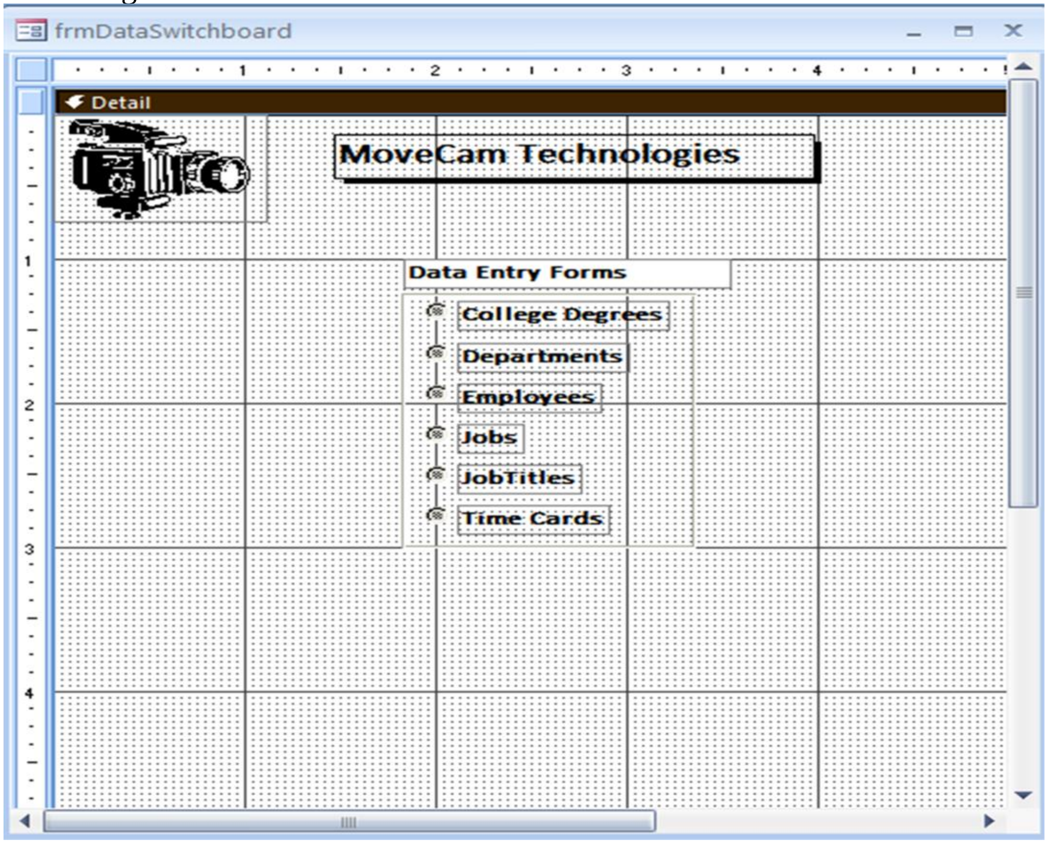
**Step 3**
Click the events tab on the properties sheet, click the **After Update** text box, click its list arrow, and then click **mfrmDataSwitchboard**. This is the name of the macro group you have just created.

**Step 4**
Click the **form selector** to display the forms properties, click the **All** tab, change the Caption property to **Forms Switchboard**, and close the property sheet.



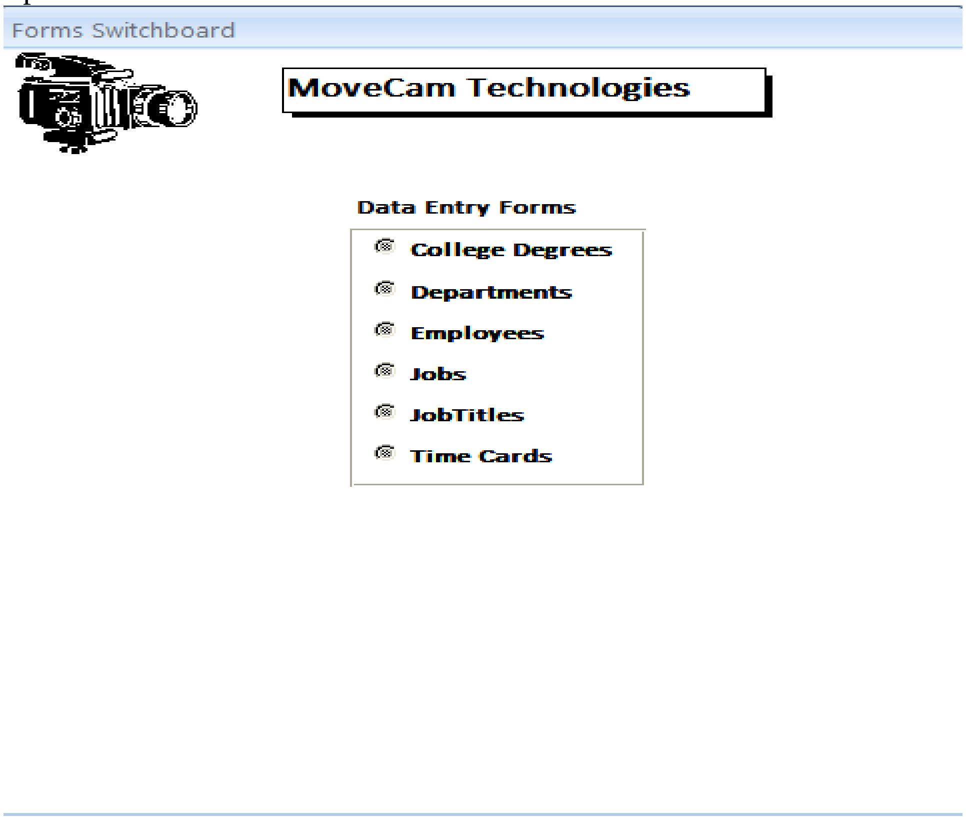
**Step 5**
Resize the form so it is 5“wide. Open the property sheet and set popup to yes. Set close button to YES. (for now you can rt-click to close the form)

**Step 6**
Move the option group and its label so that they are centered below the MovieCam Technologies label as shown below



**Step 7**
Save the changes and close the form.

**Step 8**
Open the form in form view. It should look like the form shown below:



**Step 9**
Click that **College Degrees** option button to test the macro. The College Degrees Data Entry form should appear in form view.

**Step 10**
Close the form and return to the switchboard.

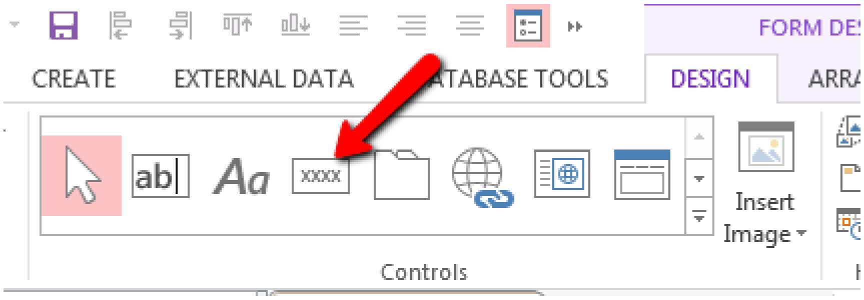
**Step 11**
To verify all forms will open use the switchboard select each form from the switchboard and then close the form.

Because the switchboard does not have a Close button in the upper-right corner, you want to add a command button that users can click to close the frmDataSwitchboard form so they will not have to use the File menu to close the form. You now will create that macro as part of a new macro group, and then assign that macro to the On Click Event property of the new button.

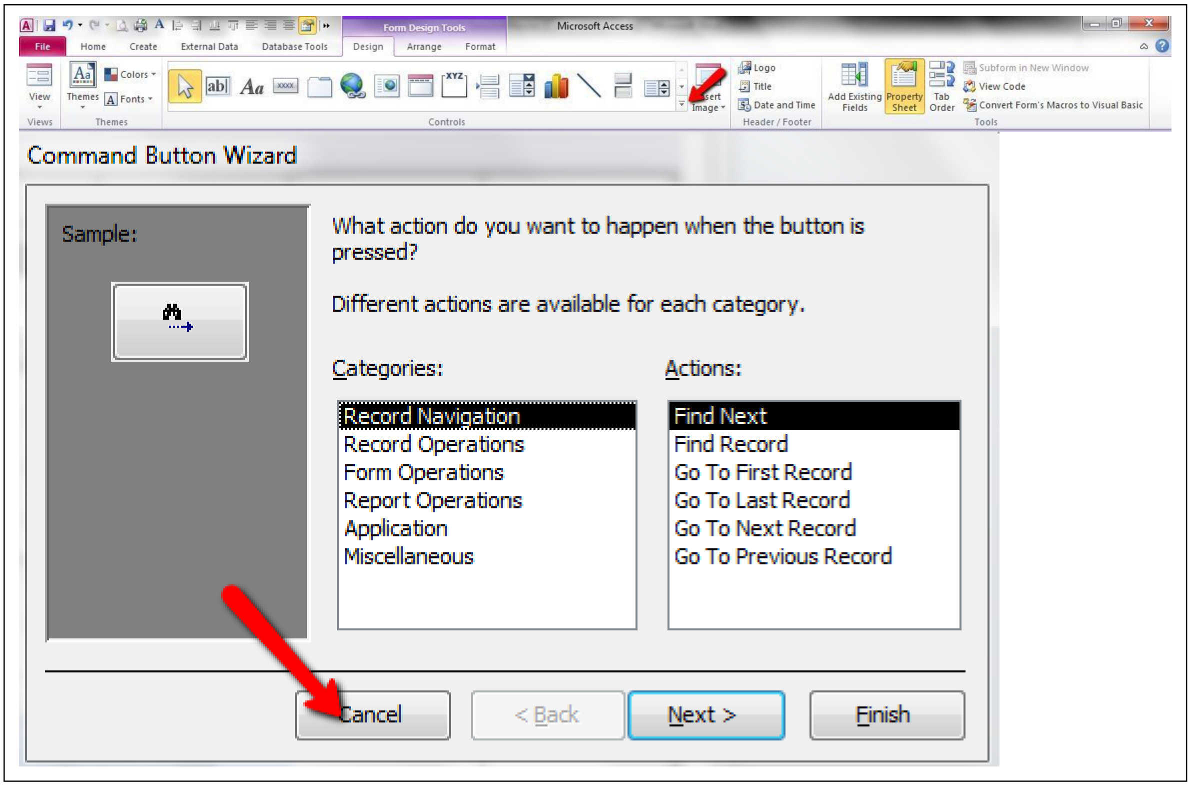
## Creating a Macro to Close Objects:

**Step 1**
Open a **frmDataSwitchboard** form in design view, and then scroll down to view the detail section to make sure that there's enough room under the option group to accommodate the new button you need to add to the switchboard. If not, increase the height of the detail section.

**Step 2**
Select the **Design** tab in the Ribbon and click the **Button** in the **Controls** group. Click in the **Detail** section and draw a command button**, (s**ee below). When the Command Button Wizard appears click the Cancel button.



**Note:** If the pop-up window does not appear it is because the Control Wizard button is not selected. This button is located in the Design Ribbon under the Controls section.



Command Button Wizard

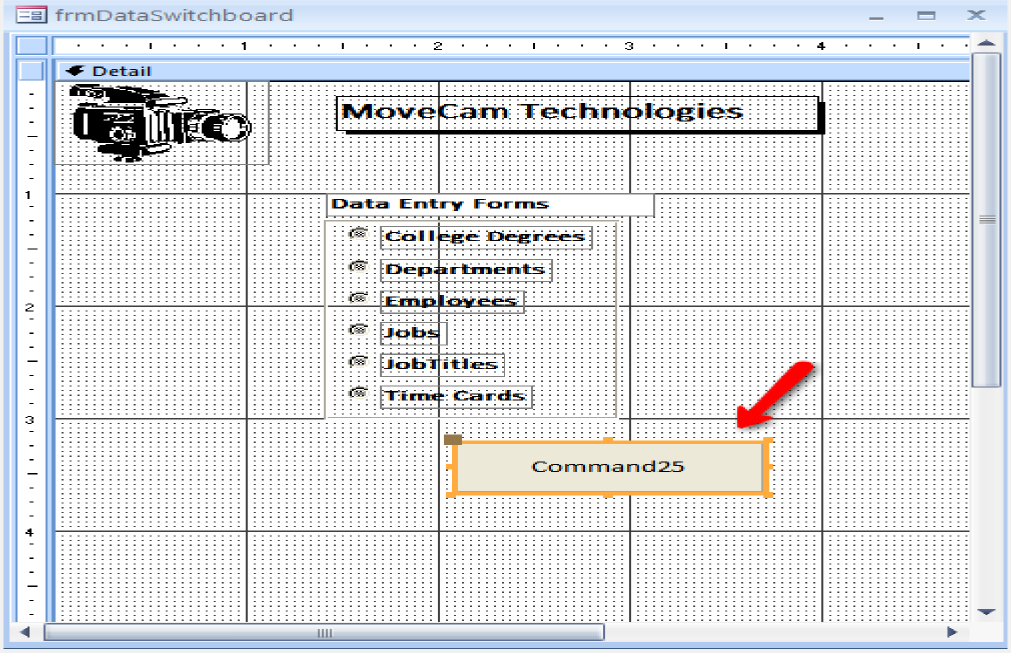
What action do you want to happen when the button is pressed?

Different actions are available for each category.

Categories:
- Record Navigation
- Record Operations
- Form Operations
- Report Operations
- Application
- Miscellaneous

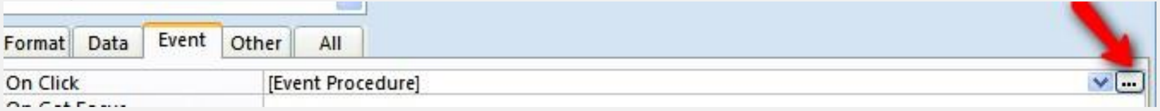
Actions:
- Find Next
- Find Record
- Go To First Record
- Go To Last Record
- Go To Next Record
- Go To Previous Record

Don't worry about the name on the current label yours may be different and it will be changed later.

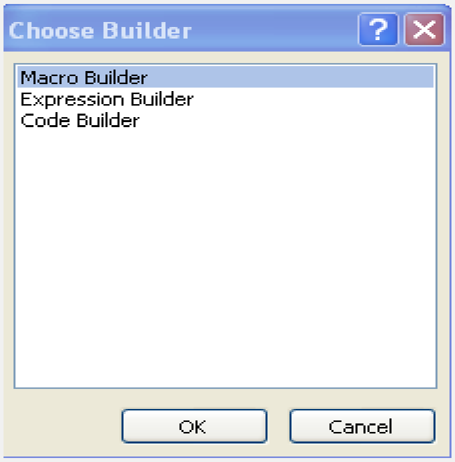


**Step 4**
Click the **Properties** button in the Ribbon, and then click the **Event** tab on the Properties Sheet:

**Step 5**
Click the **On Click** text box, then click the **Builder Button** (…). The **Choose Builder** dialog box opens (see below)

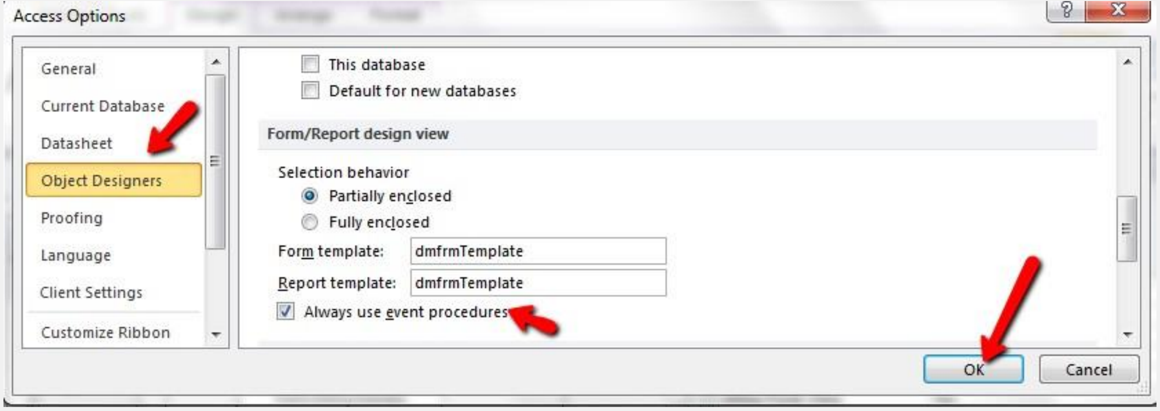


On Click [Event Procedure]



**Problem?** If the visual basic window opens automatically when you click the Build button, it is because the Always use event procedures option is checked. To disable this option, **close** the **Visual Basic Window**, close the **property sheet**, **click** on the **File** tab

**click Access Options**, in the **Object designers** section, **deselect** the **Always Use Event Procedure** option and then **click** the **OK** button. Open the property sheet, delete any text that might appear in the On Click text box and then click its Build button.



Access Options

- General
- Current Database
- Datasheet
- Object Designers
- Proofing
- Language
- Client Settings
- Customize Ribbon

□ This database
□ Default for new databases

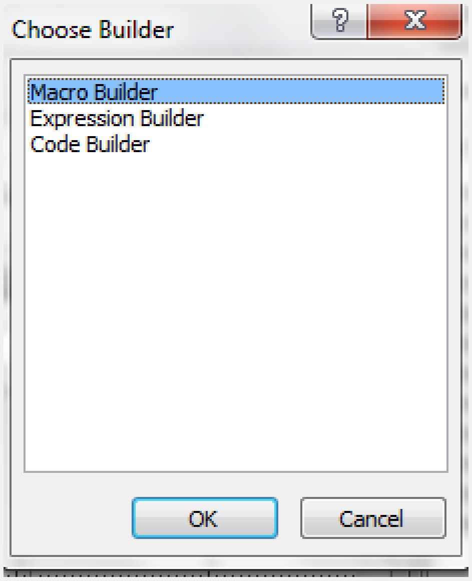
Form/Report design view

Selection behavior
● Partially enclosed
○ Fully enclosed

Form template: dmfrmTemplate
Report template: dmfrmTemplate
☑ Always use event procedures

**Step 6**

Click **Macro Builder** in the Choose Builder dialog box and then click the **OK** button.

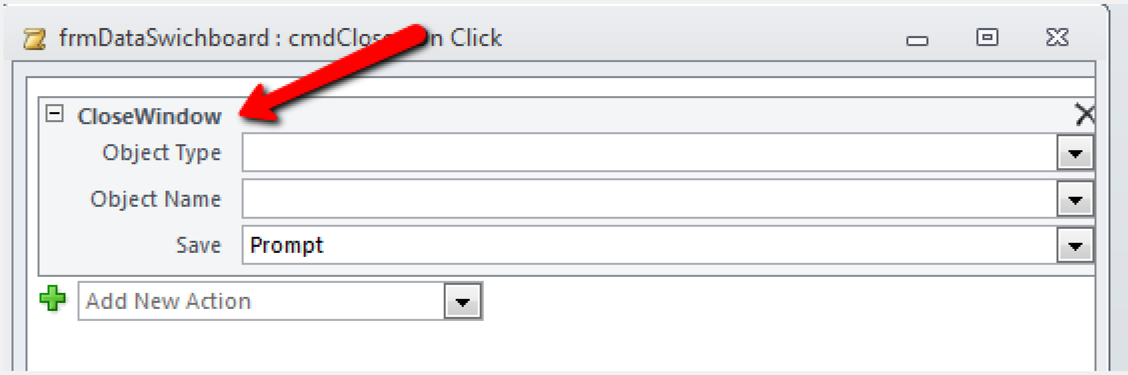


Choose Builder

- Macro Builder
- Expression Builder
- Code Builder

**Step** 7

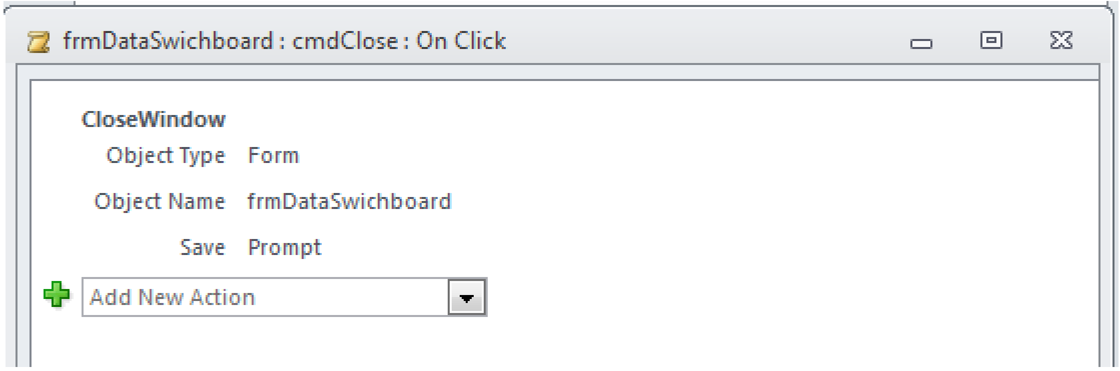
In the Add Action Box select **CloseWindow.**



frmDataSwichboard : cmdClos... n Click

CloseWindow
Object Type:
Object Name:
Save: Prompt

**Step 8**
Select **Object type** “**Form**“and **Object Name “frmdataSwitchboard”**.



frmDataSwichboard : cmdClose : On Click

CloseWindow
Object Type Form
Object Name frmDataSwichboard
Save Prompt

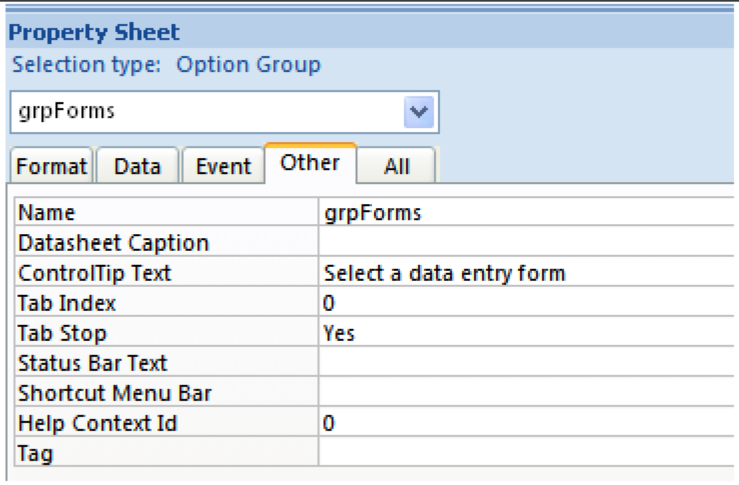
**Step 10**
Save your changes and the **close** the Macro **window**. The Property Sheet should still be open, although it may not be active.

**Completing the data in switchboard form:**
**Step 1**
If necessary, **click** the **Command button** control to select it, click the **All** tab in the properties sheet.

**Step 2**
Change the name property to **cmdClose**, and then change the caption property to **&Close**. The ampersand indicates that the letter "C." is underscored on the button, so the user can press a keyboard shortcut of **Alt + C** to execute the action and not have to click the button. If you want the ampersand character to be visible in the label or a command button, you must type two ampersands (&&). Otherwise, only the character will be visible.

**Step 3**
Click the **Other** tab, click the **Control Tip Text** text box, and then type **Return to main switchboard**. (Note that the control tip text property specifies the screen tip for the control.)

**Step 4**
Click the **Option Group Frame** to display its property in the property sheet, click in the **Control Tip Text** text box, type **Select a data entry form**, as shown below



Property Sheet

Selection type: Option Group

grpForms

| | |
|---|---|
| Name | grpForms |
| Datasheet Caption | |
| ControlTip Text | Select a data entry form |
| Tab Index | 0 |
| Tab Stop | Yes |
| Status Bar Text | |
| Shortcut Menu Bar | |
| Help Context Id | 0 |
| Tag | |

**Step 5**
Position the cmdClose button so that it is centered below the option group.

**Step 6**
**Close** the property sheet, **save** your changes, and then **switch** to Form view.

**Step 7**
Position the pointer on the center of the option group so that the ScreenTip text is visible. Be sure that you leave the pointer stationary so that the ScreenTip will appear (this might take a few seconds)

**Step 8**
Click the **close** button or press **Alt + C** which will close the form.

**Step 9**
**Exit** Access. Make certain you've added your name to the file and submit for grading.