Using Microsoft Office 2007
Advanced Access Handout

INFORMATION TECHNOLOGY SERVICES
California State University, Los Angeles

Version 2.0 Summer 2009

Contents

Advanced Microsoft Access 2007

This handout is a continuation to the Introduction to Access 2007 and Intermediate Access 2007 handout. Both handouts cover basic and intermediate functions of the **Access** program including creating various objects, modifying existing objects, creating relationships among tables and queries, working with table analyzer and features of advanced query, etc. To obtain copies, go online to http://www.calstatela.edu/its/docs/office.php. In this part the user will learn how to create other objects, such as pages and macros, produce advanced forms and charts (e.g., PivotTable and PivotChart, SubForm or SubChart), and use other advanced features. By following the instructions in this handout, it is expected that users will be able to manage their data set more effectively and efficiently using these advanced functions provided herein.

Downloading the Data Files

This handout includes sample data files that can be used for hands-on practice. The data files are stored in a self-extracting archive. The archive must be downloaded and executed in order to extract the data files.

The data files used with this handout are available for download at http://www.calstatela.edu/its/training/datafiles/access2007p3.exe.

Instructions on how to download and extract the data files are available at http://www.calstatela.edu/its/docs/download.php.

Working with Macros

A macro performs a set of commands in sequence. When used in word processors and spreadsheets, macros duplicate keystrokes or mouse movements. Macros in Access often automate an action or a series of actions. Such actions include opening tables, printing forms, finding records, or applying filters. Macros can even be used to add command buttons, create menus and toolbars, and build complete applications.

Macro commands in Access consist of an action and its arguments. The action is the task to be performed, such as opening a form. The arguments determine the specifics for the action, such as which form to open.

A macro is created from the ***Macro Builder*** which has two sections. The upper section, called the **Action** section, contains a design grid. The design grid can contain up to five columns. The ***Action*** and ***Comment*** columns are always displayed. The ***Action*** column contains one of the many available macro commands. In the ***Comment*** column, a description of the action can be typed. Comments are helpful when editing macros that contain many actions. The ***Macro Name***, ***Condition***, and ***Arguments***, columns can also be displayed. The ***Macro Name*** column contains a name for a macro that can be referred to during events, such as clicking a command button. A macro can be executed conditionally by adding a statement to the ***Condition*** column. The lower pane, called the **Action Arguments** section, contains the arguments. The arguments listed will change depending on the action selected. Arguments are values used by the macro to perform the specified actions. Most actions require one or more arguments.

The ***Arguments*** column is a new feature in Access 2007. As arguments are set in the **Action Arguments** pane for an action, they will appear in the ***Arguments*** column. The example shown in Figure 1 shows the arguments for three different actions. If an argument is not set for an

action, then only commas will appear in the Arguments column. The second action in Figure 1 has no arguments set, so the ***Arguments*** column contains only commas.

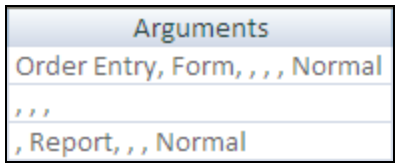

Figure 1 – Arguments Column

OPENING THE MACRO BUILDER

The design grid in the *Action* section of the ***Macro Builder*** contains at least two columns: ***Action*** and ***Comment*** (Figure 2). In the ***Action*** column, the actions to be performed by the macro are entered. The user can either type the name of the action or select one from the drop-down list box. In the ***Comment*** column, the purpose of the action is described. Adding a comment may not seem important when a macro is first created, but it can prove helpful if one needs to edit the macro at a later time.

CREATING AND SAVING A MACRO

When a field is selected in the ***Action*** column, a list of actions appears. Most of the actions are self-explanatory and have equivalent menu commands. For instance, the ***OpenQuery*** action opens a query in **Datasheet** or **Design** view depending on how the arguments are set.

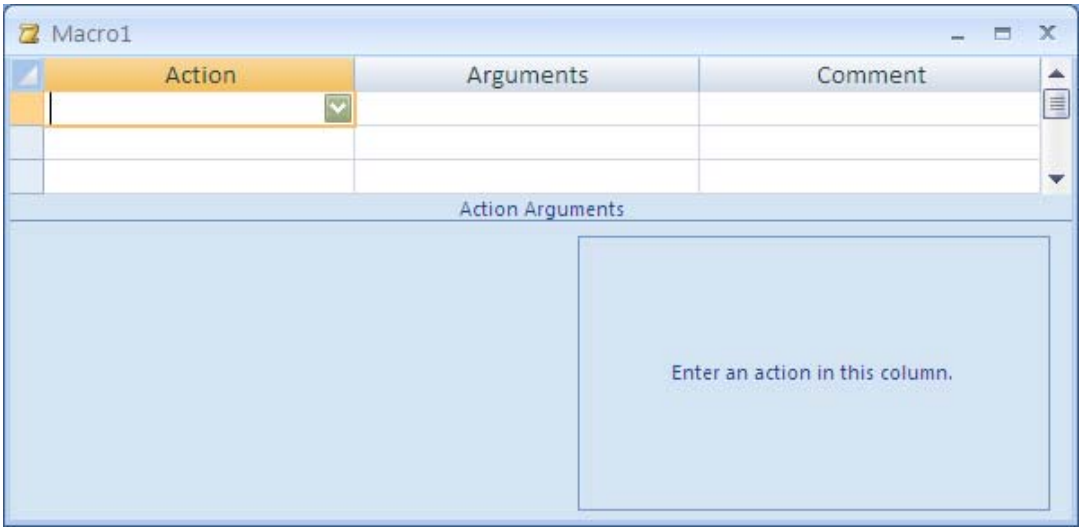

Figure 2 – The Macro Builder

A macro can include up to 999 actions. Each action is placed in a separate row in the ***Macro Builder*** in the order in which they are to be performed. For example, a ***Maximize*** command maximizes the window opened in the step immediately preceding it.

The ***Comment*** column is used to enter a description of the action to be performed. This field can contain up to 255 characters. While comments are optional, it is beneficial to enter a description of the action. This field is helpful if the macro has to be modified at a later time.

Figure 3 – Macro Drop-down Menu

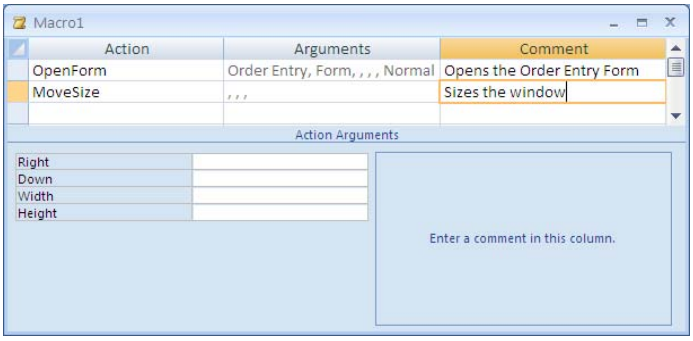

Figure 4 – Adding an Action and a Comment to a Macro

To create and save a macro:

1. Open the "***World34.accdb***" file.
2. Select the *Create* tab on the **Ribbon**.
3. Click the ***Macro*** drop-down arrow in the ***Other*** group.
4. Click the Macro button in the drop-down menu (see Figure 3). The ***Macro Builder*** opens.
5. Click in the first blank field in the ***Action*** column, if necessary.
6. Click the ***Action*** drop-down arrow.
7. Scroll as necessary and select the "***OpenForm***" command.
8. Click the ***Form Name*** drop-down arrow in the **Action Arguments** pane.
9. Select the "***Order Entry***" form.
10. Click the corresponding field in the ***Comment*** column for the "***OpenForm***" action.
11. Type **[Opens the Order Entry Form]**.
12. Add the "***MoveSize***" action to the second row of the ***Action*** column.
13. Enter the text **[Sizes the window]** in the corresponding field in the ***Comment*** column (see Figure 4).
14. In the ***Action Arguments*** pane, enter "***2***" in all arguments except for ***Width*** which should be set to "***8***."
15. Click the **Save** button on the **Quick Access Toolbar**. The *Save As* dialog box opens.
16. Enter a name into the ***Macro name:*** text box ► **OK** button

NOTE: The **[Tab]** key and the **[Shift+Tab]** key combination can be used to navigate between the columns of the **Macro Design** window.

To show or hide columns in the Macro Design window:

1. Select the *Design* contextual tab on the **Ribbon**.
2. Click either the **Macro Names**, **Conditions**, or **Arguments** button in the ***Show/Hide*** group to display or hide additional columns (see Figure 5).

Figure 5 – Show/Hide Group

DELETING AND ADDING ACTIONS FROM A MACRO

Actions can easily be removed from a macro if they are no longer required. Right-clicking the Row Select button on the left side of the argument will select the argument and open a pop-up menu. Selecting Delete Rows from the menu will remove the argument. Actions are added the same way except that the Insert Rows option is selected from the pop-up menu. The new row is added above the row that was selected.

NOTE: The same commands can also be found on the *Design* contextual tab on the **Ribbon** in the ***Rows*** group.

ASSIGNING ARGUMENTS TO AN ACTION

Most action arguments have a default list of available arguments. For example, the ***View*** argument field for the ***OpenReport*** action contains a list with the ***Print***, ***Design***, ***Print Preview***, ***Report***, and ***Layout*** arguments. For action arguments without a list, the argument can be manually typed into the argument field. A total of 255 characters can be entered in the argument field.

Some arguments are required. For example, the ***Form Name*** argument must be selected for the ***OpenForm*** action. Other arguments, such as ***Filter Name***, are not required for the ***OpenForm*** action. If a required argument is missing, the macro stops when the action containing the missing argument is encountered.

In some cases, a default argument is used. For example, the ***View*** argument defaults to ***Form*** for the ***OpenForm*** action. Other arguments are ignored if they are not selected. The ***Filter Name*** argument allows the user to select a query to apply to the form as a filter. If a query name is not entered, all the records appear.

With forms, a ***Where Condition*** argument can also be specified. This argument acts as a filter, without using an actual query. For example, the argument "**[Orders]![Shipping Method]="DHL"**" will only present records where the shipping method is "***DHL***." The condition can be typed directly into the argument text box or by clicking the **Expression Builder** button ... to open the *Expression Builder* dialog box.

NOTE: In the above example, the **[Orders]![Shipping Method]="DHL"** statement is built using Structured Query Language (SQL) syntax where "***Orders***" represents the table where the information is taken from and "***Shipping Method***" is the field in the "Orders" table.

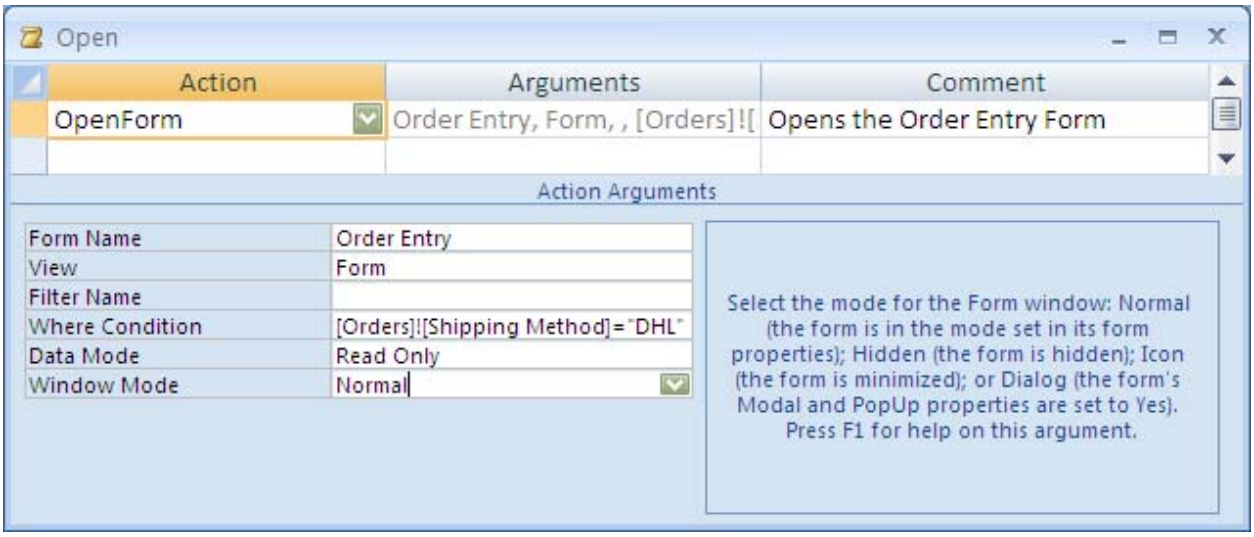

Figure 6 – Assigning an Argument to an Action

Access displays a helpful message explaining the selected argument in the right portion of the *Action Arguments* section (see Figure 6). Pressing the **[F1]** key while the insertion point is in the *Action Arguments* section of the ***Macro Builder*** will activate the Access help system.

To assign an argument to an action:

1. If necessary, open the ***Macro Builder***.
2. Select the ***OpenForm*** action.
3. Click in the ***Form Name*** argument field in the **Action Arguments** section (Figure 6).
4. Click the ***Form Name*** drop-down arrow .
5. Select the "***Order Entry***" argument.
6. Set the ***View*** argument to "***Form***," the ***Data Mode*** argument to "***Read Only***" and the argument ***Window Mode*** to "***Normal***."

   NOTE: The **[F6]** key can be pressed to toggle between the **Action** section and the **Action Arguments** sections. Also, the **[Tab]** key and the **[Shift+Tab]** key combination can be used to navigate through the argument fields.

SAVING A MACRO

After creating a macro, it must be saved before testing or running. When users attempt to run or close the ***Macro Builder*** without saving, a prompting message box will open (see Figure 7). Clicking the **Yes** button will open the *Save As* dialog box (see Figure 8) where the user can name the macro in the ***Macro Name:*** text box and then click the **OK** button.

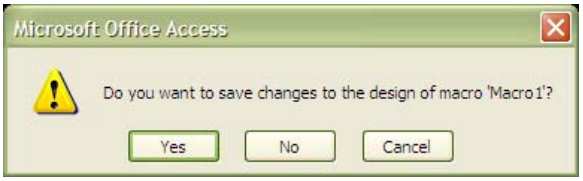

Figure 7 – Save Macro Prompt

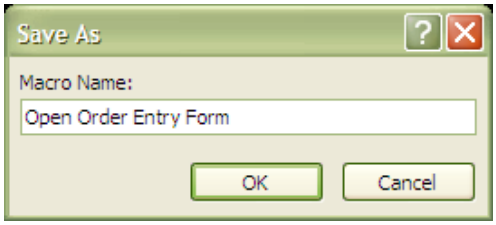

Figure 8 – Save As Dialog Box

NOTE: For this example, the user should name the macro "***Open Order Entry Form***."

The name that is assigned to the macro appears in the ***Macros*** object list in the **Objects Navigation Pane** along with all other macros associated with the database. Once a macro has been assigned a name, the user can save changes by clicking the **Save** button on the **Quick Access Toolbar**.

RUNNING A MACRO

There are several ways to run a macro depending on whether the macro is open for editing or closed.

To run a macro from the Access interface when the macro is closed:

1. Select the *Database Tools* tab on the **Ribbon**.
2. Click the Run Macro button in the ***Macros*** group. The *Run Macro* dialog box opens (see Figure 9).

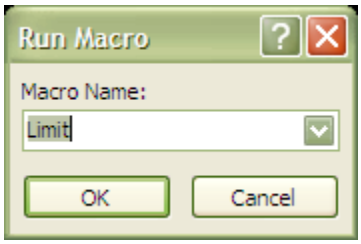

Figure 9 – Run Macro Dialog Box

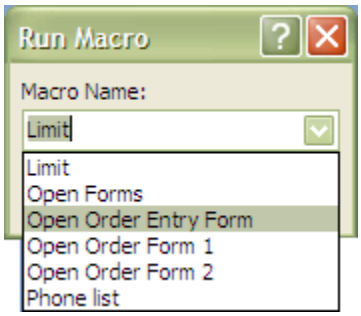

Figure 10 – Macro Name Drop-down List

3. Click the ***Macro Name:*** drop-down arrow and select the macro to run from the drop-down list (see Figure 10) ► **OK** button.

   NOTE: If the list of macros is visible in the **Object Navigation Pane**, a macro can be run by right-clicking the macro name and selecting the Run command.

To run a macro from the Access interface when the macro is in Design View:

1. Select the *Design* contextual tab on the **Ribbon**.
2. Click the Run button in the ***Tools*** group.

EDITING AN EXISTING MACRO

After a macro has been created, the user may decide to add or delete existing actions or change actions or action arguments. The user can cut, copy, paste, or type to edit a macro while the macro is open in the ***Macro Builder***. After the desired changes are made, the macro must be saved before it can be run. It is a good idea to test the edited macro using single step mode. It is important to remember to disable single step mode after testing a macro.

To edit an existing macro:

1. In the **Object Navigation Pane** right-click the "***Open Order Entry Form***" macro.
2. Select the Design View option.
3. Make all necessary changes to the actions and action arguments ► **Save** button.
4. Close the "***World34.accdb***" file.

TESTING A MACRO USING SINGLE STEP MODE

It is usually prudent to test a macro before it is put into use. This is especially true if the macro will execute multiple, complex actions or if the database will be used by someone other than the person who designed it and built the macros and other database objects. The **Single Step** function allows a user to run a macro one action at a time and observe the performance and execution of all actions in the macro. Additionally, the **Single Step** function keeps a running count of any errors that occur.

To test a macro using the Single Step function:

1. Open the "***World35.accdb***" file.
2. Display the macros in the **Object Navigation Pane**.
3. Open the "***Open Order Form 2***" macro in **Design View**.
4. Select the *Design* contextual tab on the **Ribbon**.
5. Click the Single Step button in the ***Tools*** group.

   NOTE: Clicking the Single Step button a second time will take the macro out of **Single Step** mode and cause the macro to run normally.

6. Click the Run button in the ***Tools*** group. The macro will perform the first action and the *Macro Single Step* dialog box will open (see Figure 11).

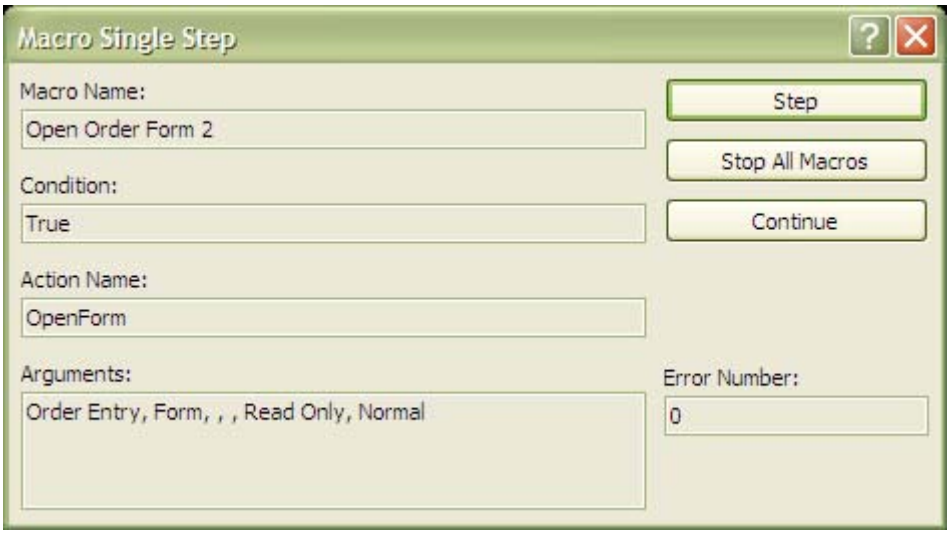

Figure 11 – Macro Single Step Dialog Box

7. Click the Step button to execute the next action in the macro.

   NOTE: Clicking the Continue button will take the macro out of **Single Step** mode and cause the macro to run normally.

8. Continue clicking the Step button until all actions have been executed
9. Close any database windows that may have opened. The macro should still be in **Design View**.
10. If necessary, select the *Design* contextual tab on the **Ribbon**.
11. Click the Single Step button to take the macro out of **Single Step** mode.

USING PROPERTIES

Properties allow the user to specify the appearance and behavior of objects in a database. Objects include tables, queries, forms, and reports, as well as controls within reports or forms.

Property Sheets display the properties of a selected object. Each has several tabs including *Format*, *Data*, *Event*, and *Other* that list the properties by group. The groups are usually the same for every property, but the items in the group change depending on the type of object selected. The *All* tab displays all the properties in a single list.

The *Format* properties allow the user to control the appearance of an object, such as color, font, size, and borders. These properties change automatically when changes are made to an object. The *Data* properties allow users to specify the source of the data and control items, such as default values. The *Event* properties allow to control when an action occurs. The *Other* properties contain items that do not fit into the other three categories.

ASSIGNING A MACRO TO A CONTROL

A macro can be associated with a control on a form or report using the *Event* properties of the control. An event is an action, such as a mouse click or a change in value that can initiate a response. The macro runs automatically when the specified event involving that control occurs.

Many events involve the control having focus. Focus means that the control can receive data from mouse clicks or keyboard actions. For example, text boxes, toggle buttons, and option buttons can have focus, since they can respond to data entry from the keyboard or mouse. Only one control can have the focus at any given time.

By linking a command button or a control with a macro, the user can make commonly used functions easier. Initiating macros this way is particularly useful because it does not require the user to know or use the Access menu structure. For example, a macro with a command button on a form can be associated to print a report. The user could then print the report by clicking the command button without having to open the report or know the correct menu commands.

To assign a macro to a control:

1. Open the "***Customers***" form in **Design View**.

   NOTE: If the ***Property Sheets*** do not automatically appear, select the *Design* contextual tab on the **Ribbon** then click the **Property Sheets** button in the ***Tools*** group.

2. Click the "***Credit Limit***" text box.
3. Click the *Event* tab on the ***Property Sheet***.
4. Click the ***On Got Focus*** property.
5. Click the **On Got Focus** drop-down arrow ► select "***Limit***."
6. Save the changes then close the "***Customers***."
7. Open the "***Customers***" form again in **Form View** and click the "***Credit Limit***" field. Notice that the Credit Limit message box generated by the "***Limit***" macro opens (see Figure 12).
8. Click the **OK** button to close the message box.
9. Close the form.

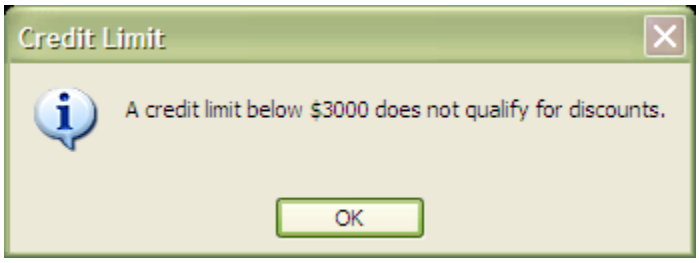

Figure 12 – Credit Limit Message Box

ADDING A CONDITION TO A MACRO

A condition argument can be added to a macro. The argument only performs the macro when the condition is met.

Before a condition can be added to a macro, the ***Condition*** column must be displayed in the ***Macro Builder***. The ***Condition*** column appears to the left of the ***Action*** column in the upper pane of the ***Macro Builder***. The condition can be typed or the *Expression Builder* dialog box can be used to create the expression.

If an expression is typed, there are certain rules that must be followed when referring to controls in tables, queries, forms, and reports. All references must be separated with an exclamation point. Additional rules for the database objects and controls are listed in the following table:

Table 1 – Reference Rules

Controls	Rules
Controls in tables	Enclose the name of the table and the name of the control in square brackets and separate them with an exclamation point. For example, **[*Orders*]![*Customer ID*]** refers to the ***Customer ID*** field in the ***Orders*** table.
Controls in queries	Enclose the name of the query and the name of the control in square brackets and separate them with an exclamation point. For example, **[*Order Items*]![*Item Number*]** refers to the ***Item Number*** field in the ***Order Items*** query.
Controls in forms	Enclose the name of the form and the name of the control in square brackets and separate them with an exclamation point. Indicate that a form is referred to by beginning the statement with the word "***Forms***". For example, **[*Forms*]![*Customers*]![*Customer ID*]** refers to the ***Customer ID*** field in the ***Customers*** form.
Controls in reports	Enclose the name of the report and the name of the control in square brackets and separate them with an exclamation point. Indicate that a report is referred to by beginning the statement with the word "***Reports***". For example, **[*Reports*]![*Customer Sales*]![*Contact Name*]** refers to the ***Contact Name*** field in the ***Customer Sales*** report.

Since the condition is entered in the ***Condition*** column in the *Macro Design* window, the word "IF" does not need to be included in the statement. Access assumes the statement is a condition. If the condition is true, Access performs the corresponding action in that row. If the condition is false, Access does not perform the action.

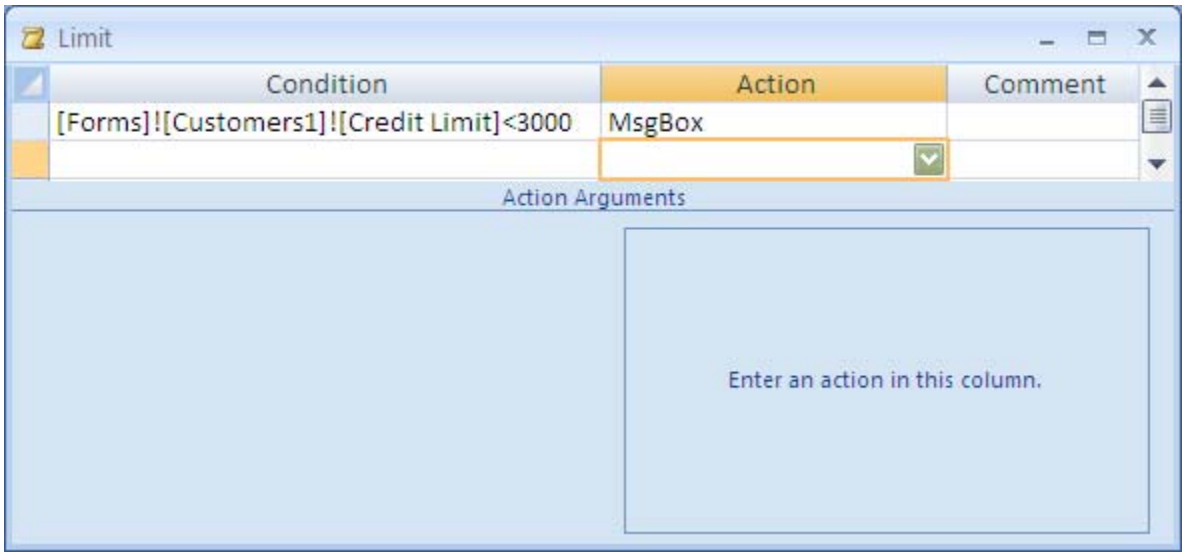

Figure 13 – Adding a Condition to a Macro

NOTE: If the condition is true, Access can be made to perform more than one action by entering an ellipsis (...) in the **Condition** column in the Macro Design window. Access performs the action on the same row as the condition and all rows thereafter that contain an ellipsis.

To add a condition to a macro:

1. Open the "***Limit***" macro in **Design View**.
2. Select the Design contextual tab on the Ribbon.
3. Click the **Conditions** button in the ***Show/Hide*** group.
4. Click in the ***Condition*** column next to the "***MsgBox***" action.
5. Type **[Forms]![Customers1]![Credit Limit]<3000** (Figure 13). Expand the ***Condition*** column, if necessary.
6. Save the macro and close the *Macro Builder*.
7. Open the "***Customers1***" form and click the "***Credit Limit***" field. Notice that the message box does not open because the credit limit for this customer is over $3000.
8. Move to record "***5***". Notice that the message box opens because the credit limit for this particular customer is under "***$3000***."
9. Click the **OK** button to close the message box then close the form.

Using the PivotTable and PivotChart View

Any database form can be viewed in ***PivotTable View*** or ***PivotChart View*** using the **View** drop-down menu when the form is open. In ***PivotTable View***, large amounts of data can be summarized and analyzed in a table format. ***PivotChart View*** allows the user to display data in a graphic format. A ***PivotTable View*** or ***PivotChart View*** can be designed by dragging fields from the ***Field List*** to preset drop areas in the ***PivotTable*** or ***PivotChart*** workspace. As an alternative to dragging fields, the user can also select the desired field in the ***Field List***, select the desired drop area from the bottom of the ***Field List***, and then use the **Add to** button to add the field to that field.

In the ***Field List***, fields are listed below a corresponding fieldset. When the fieldset is expanded, the individual fields are displayed. Field names become bolded when they are added to the view workspace.

Once the table or chart is designed, fields can be moved, added, or deleted as desired. In addition, formatting can be applied to the items in ***PivotTable*** or ***PivotChart*** to enhance its

appearance. Options in the property sheet or buttons that are found on the **Formatting** (**PivotTable/PivotChart**) toolbar can be used to apply formatting.

When a ***PivotTable View*** is created, Access automatically creates a ***PivotChart View***, and vice versa. The ***PivotTable*** layout is slightly different than the ***PivotChart*** layout, however, and may require slight modification. Specifically, when a field is dragged into the "***Drop Totals or Detail Fields Here***" area in a ***PivotTable***, that field is added as a detail field which displays the detail but not a summary of the detail. Because of the nature of this field, the information will not display on the chart in the data field area. In this instance, the user will need to add that same field to the data area in ***PivotChart View***. The addition is also reflected in the ***PivotTable***. Any formatting applied to items in each view, however, is independent of the other view.

PivotTable and ***PivotChart*** views are automatically saved and updated as part of the layout of the form when the form is closed.

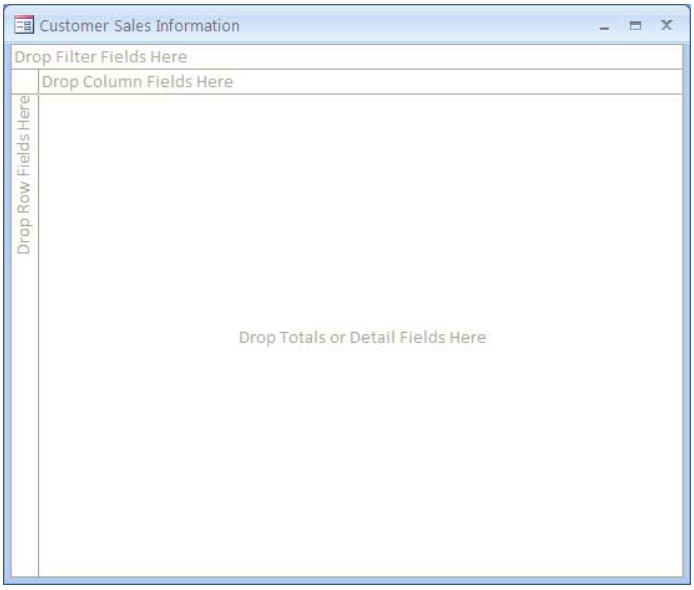

Figure 14 – Blank PivotTable

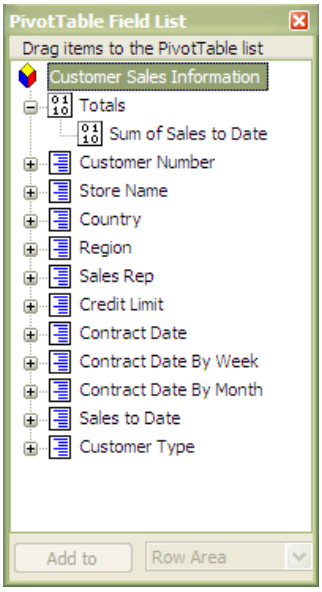

Figure 15 – Pivot Table Field List

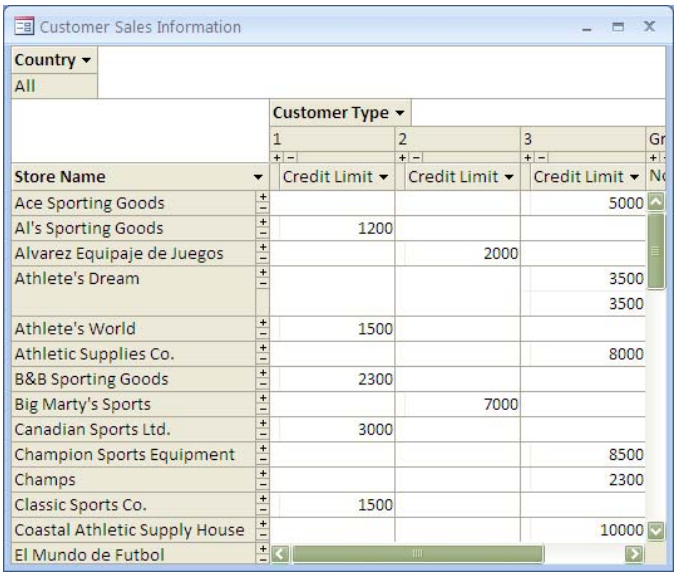

Figure 16 – Data in PivotTable

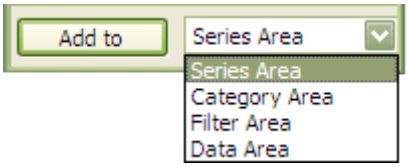

Figure 17 – Add to Drop-down List

To create a ***PivotTable*** or ***PivotChart*** view:

1. Close "***World35.accdb***" ► Open the "***World29.accdb***" file.
2. Open the "***Customer Sales Information***" form in **Design View**.
3. Select the *Design* contextual tab on the **Ribbon**.
4. Click the ***View*** drop-down arrow ► PivotTable View button. The blank ***PivotTable*** and the *PivotTable Field List* dialog box open (see Figure 14 and Figure 15).

   NOTE: If the *PivotTable Field List* dialog box does not open, click the **Field List** button in the ***Show/Hide*** group.

5. Click the ⊞ next to ***Store Name*** in the *PivotTable Field List* dialog box and drag the ***Store Name*** field to the "***Drop Row Fields Here***" area in the blank ***PivotTable*** (see Figure 14).
6. Click the ⊞ next to ***Customer Type*** in the *PivotTable Field List* dialog box and drag the ***Customer Type*** field to the "***Drop Column Fields Here***" area in the blank ***PivotTable*** (see Figure 14).
7. Click the ⊞ next to ***Sales to Date*** in the *PivotTable Field List* dialog box and select the ***Sales to Date*** field.
8. Click the ***Add to*** drop-down arrow and select "***Data Area***" from the drop-down list (see Figure 17).
9. Click the **Add to** button to add the ***Sales to Date*** field to the ***Data Area***.

   NOTE: The field selected for the ***Drop Totals*** or ***Detail Fields Here*** area of the ***PivotTable*** will not display if it is dragged. It must be added using the **Add** to button.

10. Click the ⊞ next to ***Country*** in the *PivotTable Field List* dialog box and drag the ***Country*** field to the "***Drop Filter Fields Here***" area in the blank ***PivotTable*** (see Figure 14).
11. Switch to **PivotChart View** and notice that sales to date data are displayed for each store (see Figure 18).
12. Close the form and do not save the changes.
13. Close "***World29.accdb***" ► Open the "***Trng_Sample_Adv.accdb***" file.

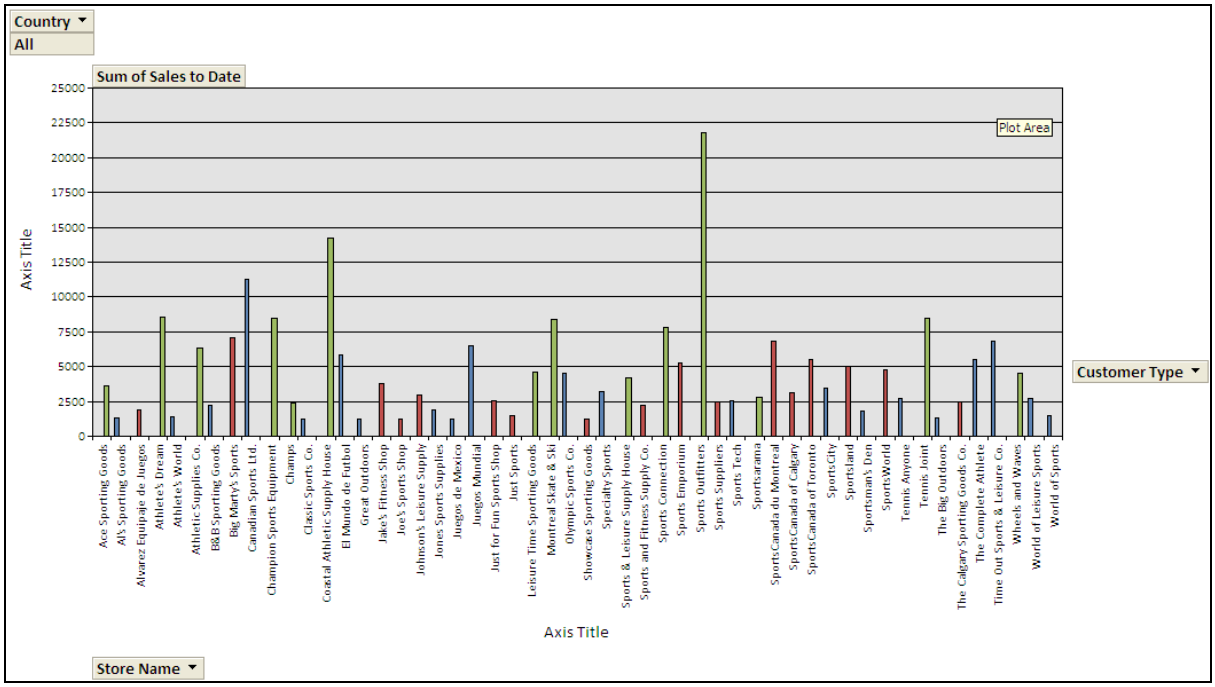

Figure 18 – Data in Pivot Chart View

Working with SubForms/SubReports

In Access it is possible to display a form/report within a form/report. The main form/report includes information from one table, and the subform/subreport includes information from a second, related table.

Subforms or ***Subreports*** are used for displaying the linked data from tables with one-to-many relationships more effectively. The main form/report and the ***Subform/Subreport*** are linked by a common field between the tables. Therefore, the main form/report represents the one side of the one-to-many relationship and the ***Subform/Subreport*** represents the many side. When viewing a record in the main form/report (one side relationship), the ***Subform/Subreport*** displays the related records from another table (many side relationship).

The ***Subform*** data can be displayed in either **Form View** (one record at a time) or **Datasheet View** (many records displayed at once), while ***Subreport*** data can only be displayed using the **Print Preview** feature. The user can also create and display ***Subform*** data in a ***PivotTable*** or ***PivotChart***.

CREATING SUBFORMS/SUBREPORTS

When adding a ***Subform/Subreport*** to an existing form, it is best to start by creating the main form/report before creating the ***Subform/Subreport***. The form/ report can then be edited in order to contain only the desired information and layout. This form can be added to the design of the main form where it will take on the properties of a ***Subform*** or ***Subreport***.

The user can create a ***Subform*** that will appear in **Form View**, **Datasheet View**, or both views. When the form is created using the ***Subform Wizard***, a ***Subform*** is created that can be viewed in both **Form View** and **Datasheet View**. The ***Subform*** can be customized in **Design View** by adding headers, footers, color, fonts, totals, etc.

The ***Subform/Subreport Wizards*** provide the easiest method for creating a ***Subform*** or ***Subreport***. If the user chooses not to use the ***SubForm/SubReport Wizards***, a form/report will be needed to be designed and then inserted into another form/report. The wizard need not be used if a ***Subform/Subreport*** is being designed that is very different in appearance from the form/ report the wizard produces. However, the ***Subform/Subreport*** makes easy not only to use an existing form/report as the subform/subreport or build a new one from tables and queries, but also to define the link between the main form/report and subform/subreport or choose from a list of available link options.

For instance, both "***Contacts***" and "***Enrollments***" tables are related with common field, "***SSN***." The "***Contacts_Report***" is created from the "***Contacts***" table. A ***Subreport*** may be created in "***Contacts_Report***" to display each individual's enrollment within one ***Report View***.

To create a ***Subreport*** (or ***Subform***) using the wizard:

1. Open the "***Contacts_Report***" report in **Design View**.

   NOTE: When opening a form/report in **Design View**, *Report Design Tools* contextual tabs appear on the **Ribbon**. The three contextual tabs associated with this tab are the *Report*, *Design*, and *Page Setup* contextual tabs.

2. Select the *Design* contextual tab on the **Ribbon**.
3. Click the **Subform/Subreport** button  in the ***Controls*** group. Users may notice that the mouse pointer changes into the **Subform/Subreport** tool button with a plus sign.

4. Click the mouse to insert a ***Subreport*** in the ***Detail*** section of the report. The ***SubReport Wizard*** opens (see Figure 19).
5. Click the ***Use existing Tables and Queries*** option button to select original data as the basis of the ***Subreport*** ► **Next >** button. The next dialog box of the wizard opens (see Figure 20).
6. From ***Tables/Queries*** drop-down list box select the "***Enrollment***" table.
7. Transfer all fields to the selected fields using the rightward double-pointing arrow (see Figure 20) ► **Next >** button.

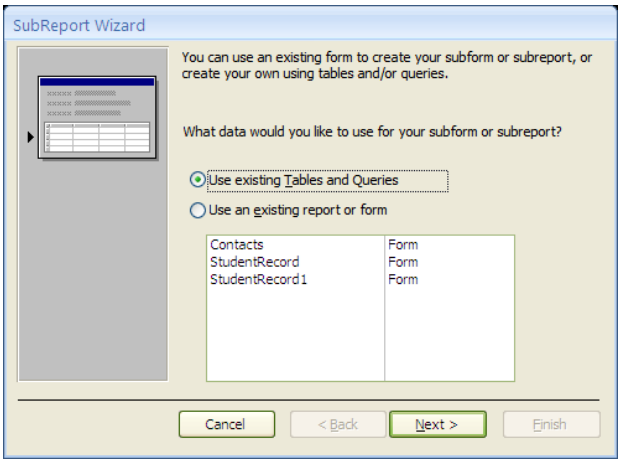

Figure 19 – SubReport Wizard: Step 1

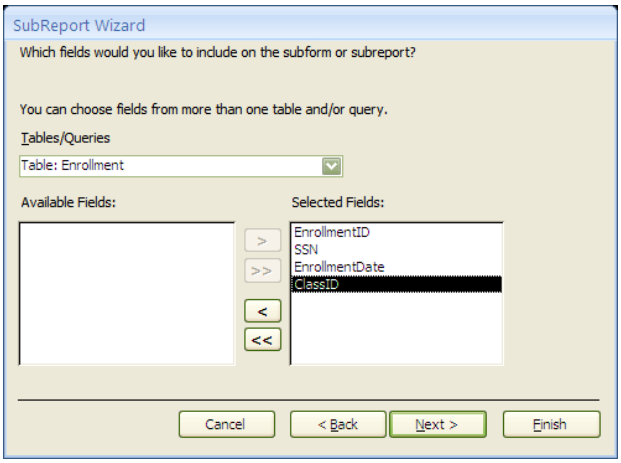

Figure 20 – SubReport Wizard: Step 2

8. Select the ***Choose from a list*** option button. Then select "***Show Enrollment for each record in Contacts using SSN***" from the text box below (see Figure 21) ► **Next >** button.
9. Name the ***Subreport*** ► **Finish** button (see Figure 22).

Figure 23 shows the "***Contacts_Report***" including the "***Enrollment***" Subreport in **Design View**. Figure 24 shows the "***Contacts_Report***" including the "***Enrollment***" Subreport in **Report View** which is very similar to **Print Preview**.

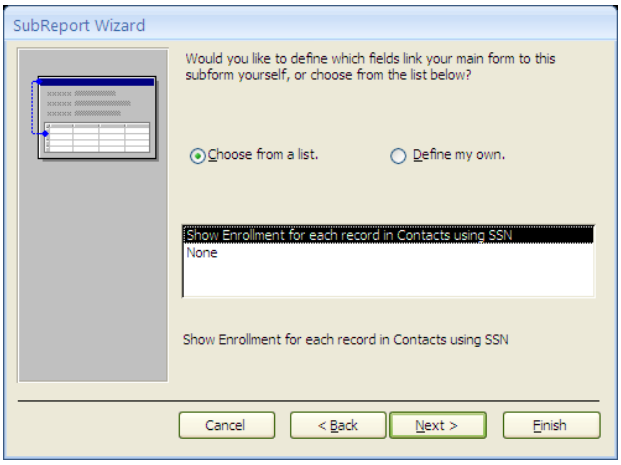

Figure 21 – SubReport Wizard: Step 3

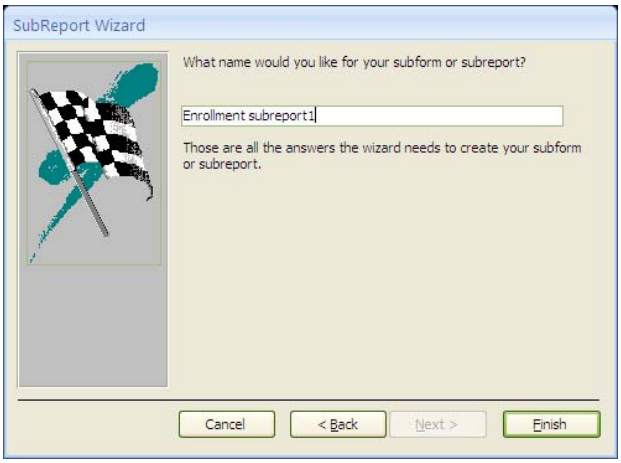

Figure 22 – SubReport Wizard: Step 4

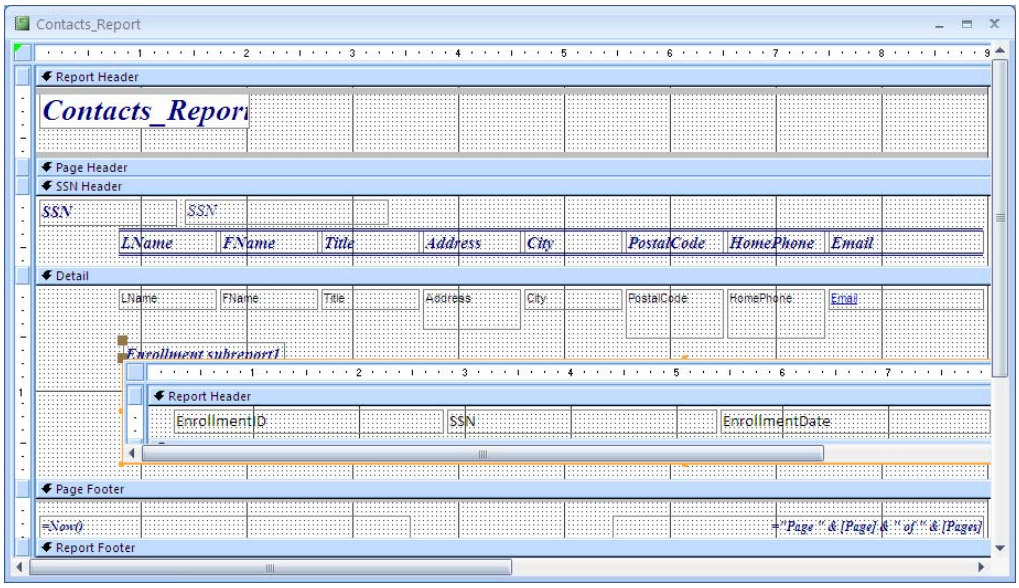

Figure 23 – Report with Subreport in Design View

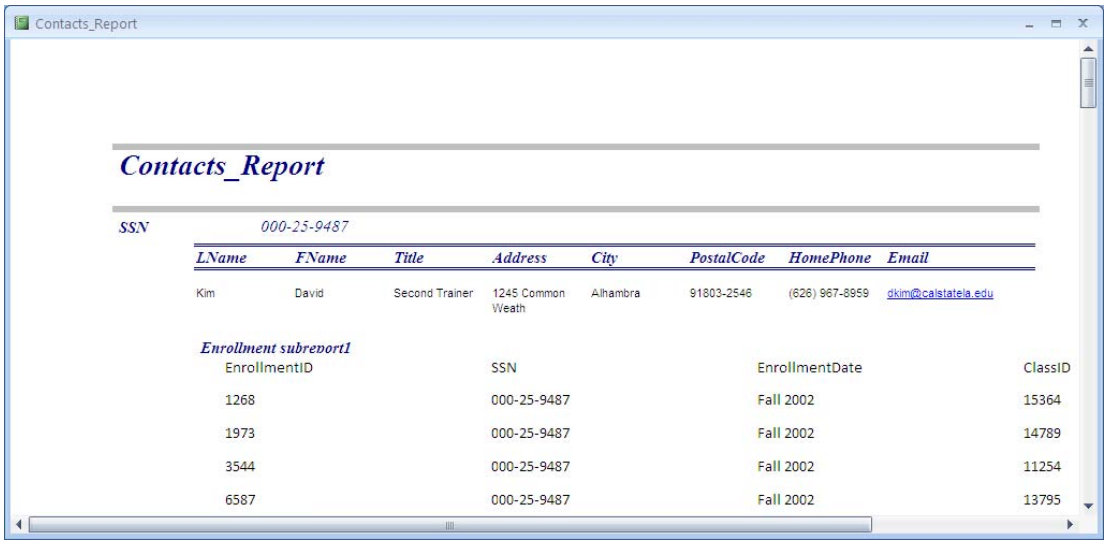

Figure 24 – Report with Subreport in Report/Print Preview View

CHANGING LAYOUT OF A SUBFORM

To improve the efficiency or appearance of a ***Subform/Subreport***, its layout can be changed using the same method to change the layout of a form/report. In **Design View** for the main form/report, the ***Subform/Subreport*** is a control that can be moved, resized, aligned, added, and deleted as would any other control. Since the ***Subform/Subreport*** control contains a form/report, it can be opened in **Design View** and any editing changes can be made to its controls. Any changes to the layout of the ***Subform/Subreport*** are done in **Design View**.

NOTE: A **Subform** can be opened into a separate **Design** window by right-clicking the desired **Subform** and then selecting the **Subform in New Window** command from the pop-up menu.

To change layout of a SubReport:

1. Open the "***Contacts_Report***" report in **Design View** (see Figure 23).
2. Double-click the ***EnrollmentId*** unbound control.
3. Double-click the ***Subform*** control text to select it.

4. Change the text to "***EroID***" (see Figure 25). When the report is run, the ***Subreport*** header will read "***EnrolID***" instead of "***EnrollmentID***."

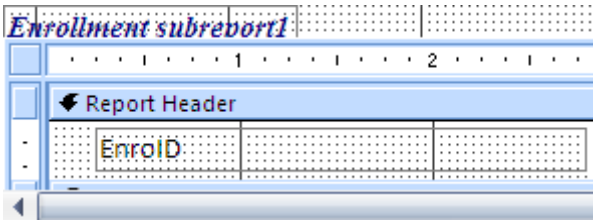

Figure 25 – Change Layout of SubReport

Generating Multicolumn Reports as Mailing Labels

Access allows printing multicolumn reports. A single column report can be created with the ***Report Wizard***, then the report can be arranged to print values from the ***Detail*** section in a specified number of columns across the page. The most common application of multi-column reports is the creation of mailing labels.

CREATING MAILING LABELS WITH LABEL WIZARD

Mailing lists can be created with the ***Label Wizard*** or can be started with a blank form. The advantage of the ***Label Wizard*** is that it includes the dimensions of virtually every kind of adhesive label for dot matrix or laser printers made by the Avery Commercial Products division and several other North American and overseas manufacturers. The user selects the product number of label that is to be used, and Access determines the number of columns, rows per page, and margins for the report's detail section. The ***Label Wizard*** can also be customized for labels with unusual sizes or those produced by manufacturers who aren't included in the repertoire of the ***Label Wizard***. Many other manufacturers include a note that indicates the corresponding Avery label number.

To create a mailing label with Label Wizard:

1. Close all open reports and select the report that the labels will be based on in the **Object Navigation Pane**.
2. Select the *Create* tab on the **Ribbon**.
3. Click the Labels button in the ***Reports*** group. The ***Labels Wizard*** opens (see Figure 26).
4. Scroll down the options in the *What size label would you like?* section until the appropriate label appears and select it. For this example, select the first item in the list ("***C2160***") ► **Next>** button.
5. Select the font options for the labels (see Figure 27) ► **Next>** button.
6. Select the fields that will be used in the labels by clicking the rightward pointing transfer arrow. A new line on the label layout can be started by pressing the **[enter]** key (see Figure 28). For this example, transfer the "***FName***" then "***LName***" fields [**enter**] key; "***Address***" field [**enter**] key; "***City***" then "***PostalCode***" fields ► **Next>** button.
7. If it is desired to have the labels sorted, transfer the sort field(s) using the transfer arrow (see Figure 29) ► **Next>** button.
8. Enter a name for the label report ► **Finish** button. The report opens in **Print Preview** (see Figure 31).

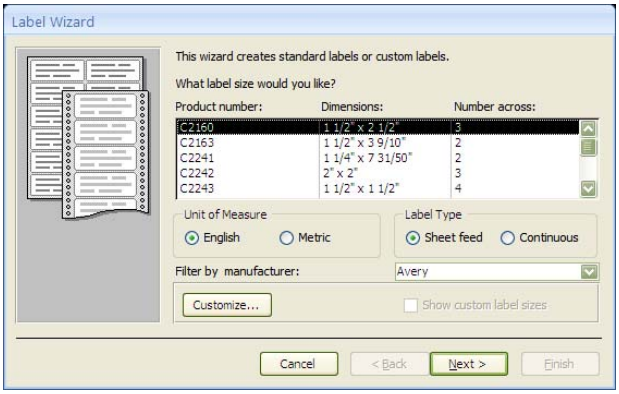

Figure 26 – Label Wizard Step 1

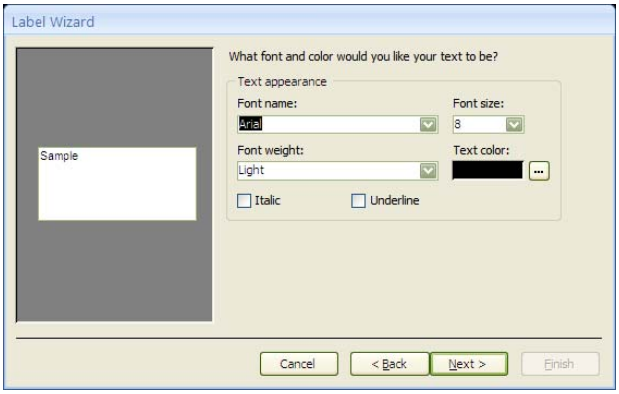

Figure 27 – Label Wizard Step 2

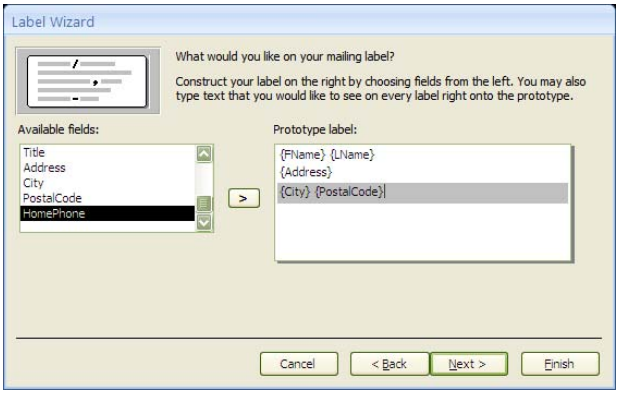

Figure 28 – Label Wizard Step 3

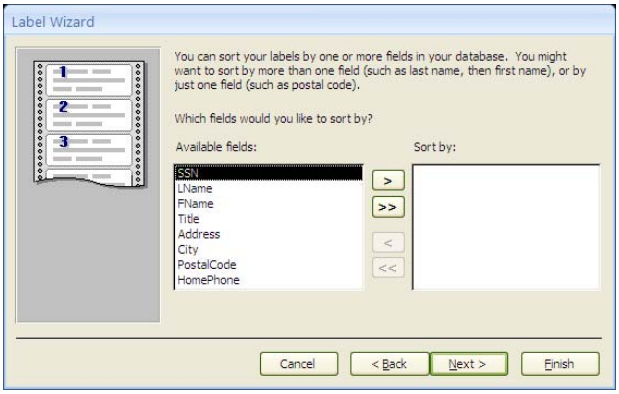

Figure 29 – Label Wizard Step 4

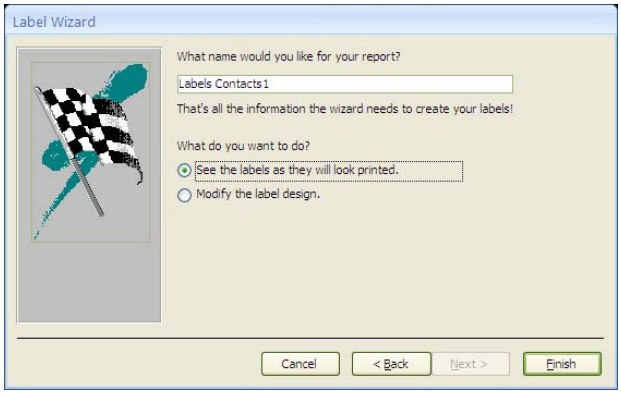

Figure 30 – Label Wizard Step 5

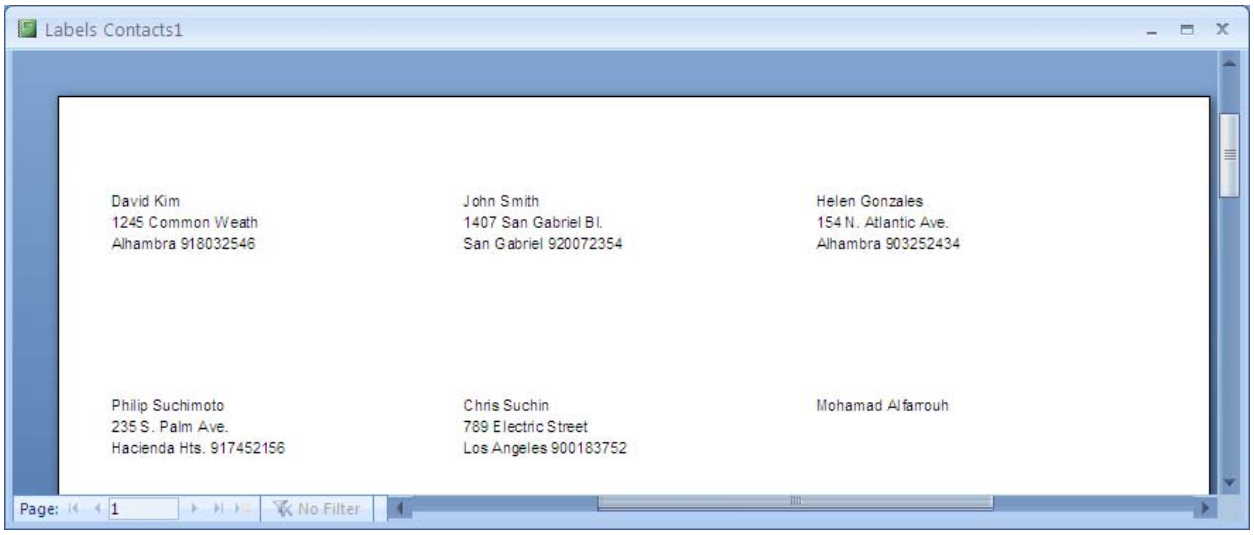

Figure 31 – Label Report in Print Preview

Using Advanced Database Features

Access contains several built-in features and utilities that assist a user to maintain optimal performance of a database file.

COMPACTING A DATABASE

When tables and other objects are deleted in a database, the database size on the disk does not necessarily decrease. Access provides a utility that compacts (defragments) the database, thereby releasing storage space. More importantly, regularly compacting and repairing a database file will help to ensure data integrity. In Access 2007, a database can be compacted and repaired while the file is open.

To compact a database:

1. Open the database file that will be compacted and repaired.
2. Click the **Office Button**.
3. Select the Manage option.
4. Select the Compact and Repair Database option.

NOTE: The user may or may not notice activity while the file is compacted and repaired.

BACKING UP A DATABASE

It is important to backup databases on a regular basis. Most corporations have a backup process to ensure that all data is saved at least once a day; consequently, all databases shared on a company network server are backed up daily.

A database may be saved to the hard drive of a personal computer because most databases are too large to fit on a floppy disk. A hard drive should be backed up regularly by saving it to another drive or to removable media (a tape or CD, for example). Backing up the hard drive ensures that all data, queries, forms and reports in your database are saved. As a result, a database can be recovered if something were to happen to it.

If neither of the above options is available, individual components (such as tables, queries, forms, etc.) can be exported to a floppy disk. In addition, data can be saved as ASCII text or as Excel tables; however, queries, forms, or reports cannot be saved in this manner.

To back up a database:

1. Open the database file.
2. Click the Office Button.
3. Select the Manage option.
4. Select the Back Up Database option. The *Save As* dialog box opens.
5. Navigate to the appropriate drive/folder using the Save in: drop-down arrow.

   NOTE: Access automatically provides a name for the backup copy of the database using the file name with the date the backup was created appended to the name. This allows the user to know when the database was backed up just by looking at the file name. It is recommended that users accept the Access name for the backup file.

6. If a different file name is desired, enter it in the File name: text box ► **Save** button.

USING NAME AUTOCORRECT

The ***Name AutoCorrect*** feature automatically fixes inconsistencies that can occur when tables, forms, reports, queries, fields, or other controls in an Access database are renamed.

As shown in Figure 32 below, the ***Name AutoCorrect*** feature has three options; ***Track name AutoCorrect info***, ***Perform name AutoCorrect***, and ***Log name AutoCorrect changes***. The ***Track name AutoCorrect info*** option saves information about changes, but does not automatically repair the inconsistencies. When the ***Perform name AutoCorrect*** option is enabled, Access automatically corrects any differences between related objects. Furthermore, if both options are enabled, the ***Log name AutoCorrect changes*** option can also be enabled, which creates a table named ***Name AutoCorrect Log*** that documents changes.

Name AutoCorrect is enabled by default for new Access 2007 databases. However, if the user is working in a converted database, it must be manually enabled.

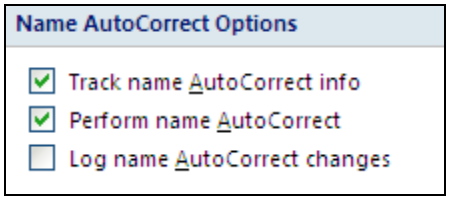

Figure 32 – Access Options Dialog Box: Name AutoCorrect Options

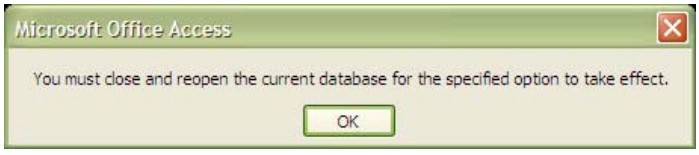

Figure 33 – Access Warning Box

To deactivate the ***Name AutoCorrect***:

1. Click the **Office Button** ► Access Options. The *Access Options* dialog box opens
2. Select the Current Database option.
3. Scroll down until the *Name AutoCorrect Options* section is visible (see Figure 32).
4. Deselect the ***Track name AutoCorrect info*** and ***Perform name AutoCorrect*** check boxes ► **OK** button.

   NOTE: If the database file where these options are deselected is open, a warning box will appear. Click the **OK** button then follow the instructions (see Figure 33).

When the ***Name AutoCorrect*** features are disabled, it cannot automatically track changes in the database.

To test this, follow steps below:

1. Open the "***Contacts***" table in **Datasheet View**.
2. Change the "***FName***" field to ***FirstName***.
3. Save the changes and close the table.
4. Open the "***Contacts***" form in **Form View**.

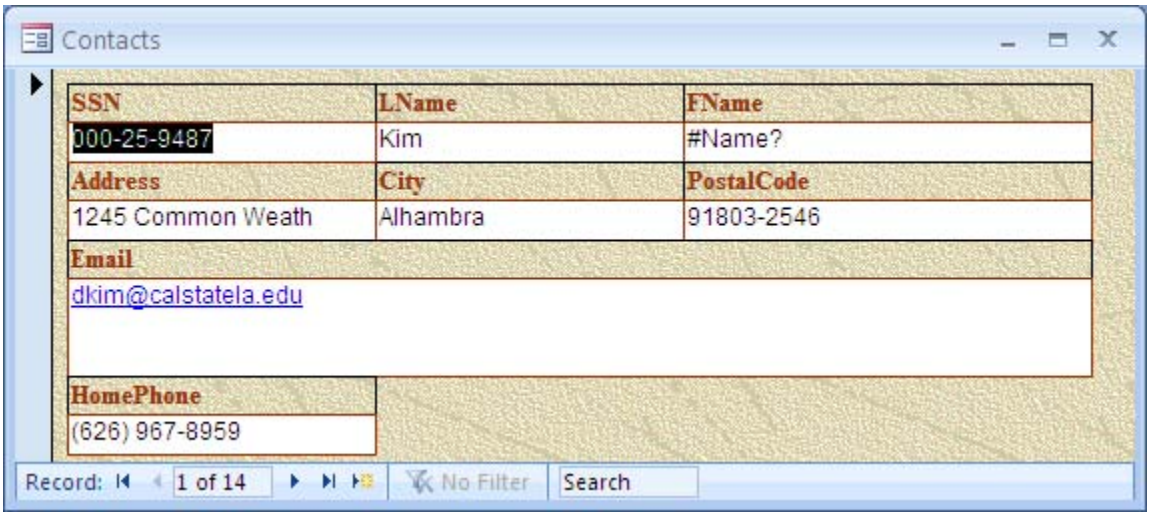

Figure 34 – Name AutoCorrect Disabled

Notice that Access is unable to find values for the ***FNname*** field from original "***Contacts***" table information. The data in the ***FName*** field on the form contains "#***Name***?" indicating that the field is no longer recognized because the name changed (see Figure 34). This can be checked in reverse situation. If the ***Name AutoCorrect*** feature is enabled, Access automatically keeps track on changes in any objects and finds correct values from earlier database to new ones.

To activate the ***Name AutoCorrect***:

1. Close the "***Contacts***" form.
2. Click the **Office Button** ► Access Options. The *Access Options* dialog box opens
3. Select the Current Database option.
4. Scroll down until the *Name AutoCorrect Options* section is visible (see Figure 32).
5. Select the ***Track name AutoCorrect info*** and ***Perform name AutoCorrect*** check boxes ► **OK** button. The same warning box appears (see Figure 33).
6. Acknowledge the warning box and follow the instructions.
7. Open the "***Contacts***" table in **Datasheet View**.
8. Change the "***Address***" field to "***Add***."
9. Save the changes and close the table.
10. Open the "***Contacts***" form in **Form View**. Notice that this time Access is able to find values for the renamed "***Address***" field (see Figure 35).

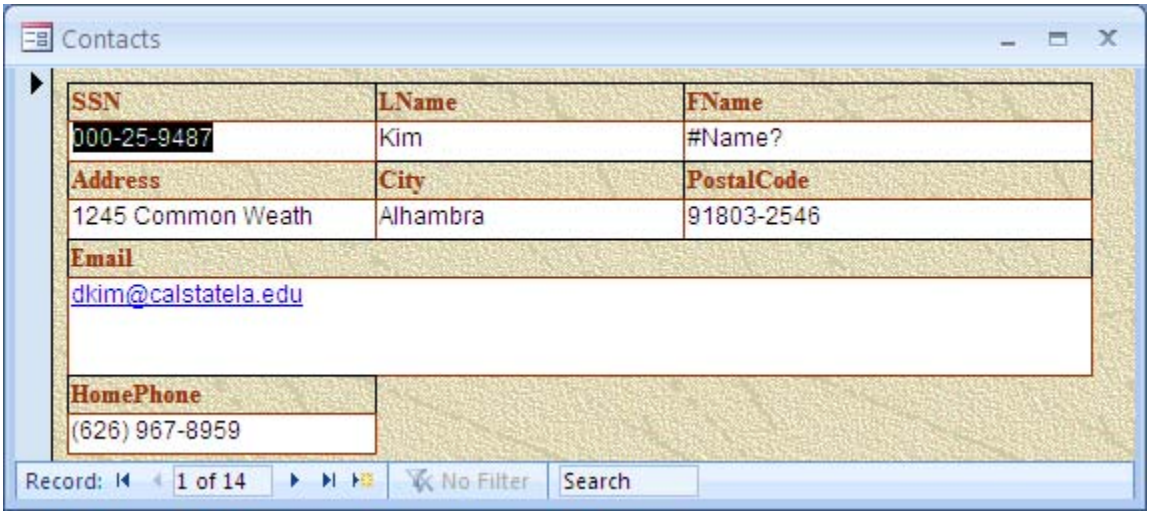

Figure 35 – Name AutoCorrect Activated

PRINTING A RELATIONSHIP DOCUMENT

The ***Documenter*** feature allows the user to view, print, and save the design characteristics of database objects, such as database structure, relationships, and information about properties associated with tables, queries, forms, and reports. It is useful when the user decides what needs to be changed or what should be maintained in its original form.

It is a good idea to document database structure and relationships. Useful information such as the properties associated with tables, queries, forms, and reports can be viewed and printed. This type of information is very useful for deciding what changes are to be made or when a database that was developed by someone else has to be maintained.

To preview the structure of relationship from current database file:

1. Select the *Database Tools* tab on the **Ribbon**.
2. Click the Database Documenter button in the ***Analyze*** group. The *Documenter* dialog box opens (see Figure 36).
3. Select the *Current Database* tab ► click the ***Relationships*** check box ► **OK** button. The diagram of relationships in the database opens in **Print Preview**.
4. Click the **Print** button in the ***Print*** group to print the diagram.
5. When finished, click the **Close Print Preview** button in the ***Close Preview*** group.

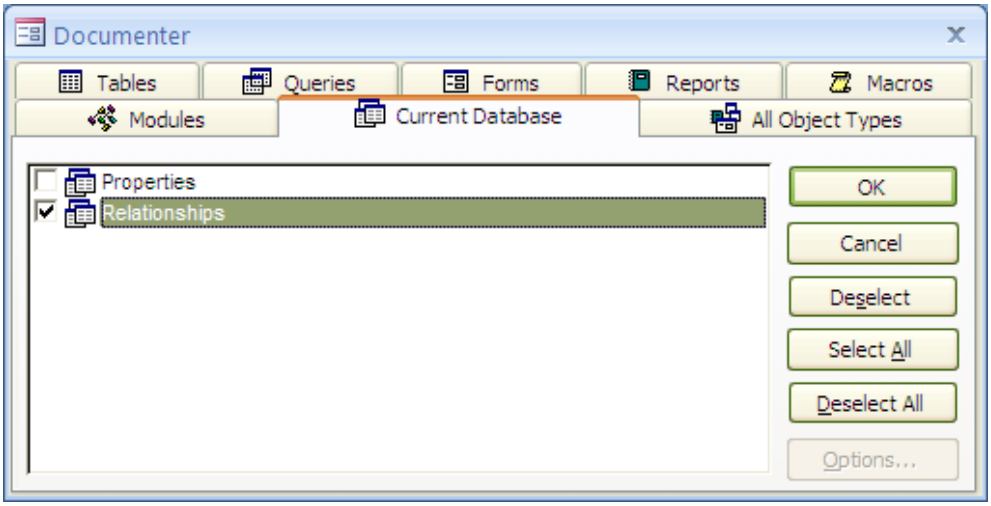

Figure 36 – Documenter Dialog Box

Appendix 1: Commonly Used Actions and Descriptions in Microsoft Access Macros

Action	Description
AddMenu	Creates custom menus under the Menus Command in the Add-Ins group.
ApplyFilter	Applies a filter or query to restrict or sort records.
Beep	Sounds a tone through the computer's speaker.
CancelEvent	Cancels the event that caused Access to run the macro containing this action.
ClearMacroError	Clears information about an error stored in the MacroError object
Close	Closes a specified window.
CloseDatabase	Closes the current database.
FindNext	Locates the next record that meets criteria specified by a find.
FindRecord	Locates the first record following the current record that meets criteria specified in the arguments.
GoToControl	Activates the specified field or control.
GoToPage	Activates the first control on a specified page.
GoToRecord	Makes the specified record the current record.
Hourglass	Changes the mouse pointer to an hourglass while the macro is running.
LockNavigationPane	Prevents users from deleting objects in the Object Navigation Pane.
Maximize	Enlarges the active window to fill the screen.
Minimize	Reduces the active window to a button on the taskbar.
MoveSize	Moves and/or resizes the active window.
NavigateTo	Controls the display of objects in the Object Navigation Pane.
OnError	Specifies what happens when an error occurs in a macro.
OpenForm	Opens a form in the selected view.
OpenQuery	Opens a query in the selected view.
OpenReport	Opens a report in the selected view.
OpenTable	Opens a table in the selected view.
OutputTo	Outputs data in the specified object to Excel text (.XLS), rich-text (.RTF), or MS-DOS text (.TXT).
Quit	Quits Access.
RemoveAllTempVars	Removes all temporary variables created using the SetTempVar action.
RemoveTempVar	Removes a single temporary variable created using the SetTempVar action.
RepaintObject	Completes any pending screen updates for a specified database object or for the active database object.
Requery	Updates the data in a specified control of an active object or updates itself if no control is specified.
Restore	Restores a maximized or minimized window to its previous size.
RunCommand	Runs a built-in Access command.
SearchForRecord	Search for a specified record in a table, query, report or form.
SelectObject	Selects an object.
SendObject	Includes the specified datasheet, form, report, module in an e-mail.

Action	Description
SetDisplayedCategories	Specifies which categories are displayed under the Navigate to Category command in the title bar of the Object Navigation Pane.
SetMenuItem	Sets the state of menu items on custom or global menus on the Add-Ins tab.
SetProperty	Sets a property for a control on a form or report.
SetTempVar	Creates a temporary variable and sets it to a specific value.
ShowAllRecords	Removes any applied filter from the applicable active object and displays all records.
SingleStep	Pauses macro execution and opens the Macro Single Step dialog box.
StopAllMacros	Stops all currently running macros.
StopMacro	Stops the currently running macro.