

Using Microsoft Office 2003 Advanced Access Handout

INFORMATION TECHNOLOGY SERVICES
California State University, Los Angeles

Version 1.1

Fall 2005

Contents

Downloading a Data File	2
Creating a Macro	2
Creating a Macro.....	3
Assigning An Argument To An Action.....	4
Saving A Macro	5
Running A Macro	6
Editing An Existing Macro	6
Creating a Macro Using the Macro Builder.....	6
Using Properties.....	7
Assigning a Macro to a Control.....	7
Adding a Condition to a Macro	8
Using Other Form Techniques	9
Creating A PivotTable Or PivotChart View	9
Using Data Access Pages	11
Creating a Data Access Page	11
Working with SubForms/SubReports.....	13
Creating SubForms/SubReport.....	14
Changing Layout of a SubForm.....	16
Generating Multicolumn Reports as Mailing Labels	16
Creating Mailing Label with Label Wizard.....	17
Creating Custom Toolbars.....	19
Create a Custom Toolbar	19
Adding a Built-in Menu Item.....	20
Creating a Custom Menu Item.....	20
Adding a Command to a Menu Item.....	21
Adding Custom Commands.....	22
Linking a Toolbar to a Report.....	23
Adding a Separator Bar.....	23
Adding Buttons to a Custom Toolbar	24
Deleting a Custom Toolbar.....	24
Using Advanced Database Features	25
Compacting a Database.....	25
Backing up a Database.....	25
Using Name Autocorrect	25
Printing a Relationship Document.....	27
Appendix 1 - Commonly Used Actions and Descriptions in MS Access Macros.....	28

For additional written instructional guides, visit us @ <http://www.calstatela.edu/handouts>

Advanced Microsoft Access 2003

This handout is a continuation to the [Introduction to Access 2003](#) and [Intermediate Access 2003](#) handouts. Both handouts cover basic and intermediate functions of the **Access** program including creation and modification of 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/handouts>. 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 a data set more effectively and efficiently utilizing the advanced functions.

Downloading a Data File

The sample data file used for the Introductory and the Intermediate handouts can be used. However, it is recommended to download the “*Trng_Sample-Adv.exe*” file for this handout by following the instructions below. Once the file is downloaded, it can be used to perform the procedures described in this handout.

To download files from the Internet:

1. Launch **Internet Explorer**.
2. Type the following address in the **Address** bar to download the file:
[\[ftp://ftp.calstatela.edu/pub/its/Trng_Sample-Adv.exe\]](ftp://ftp.calstatela.edu/pub/its/Trng_Sample-Adv.exe) ► **[Enter]**
key. The *File Download* dialog box opens (see Figure 1).
3. Click the **Save** button. The *Save As* dialog box opens (see Figure 2).

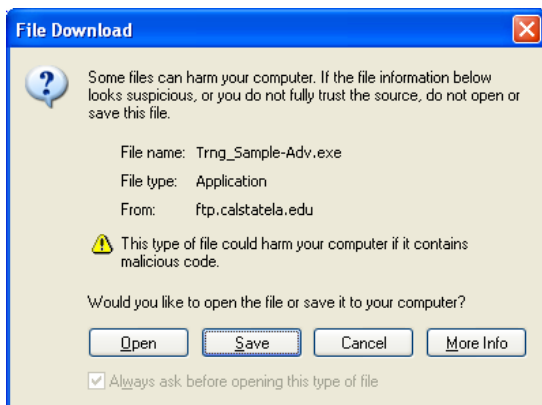


Figure 1 - File Download Dialog Box

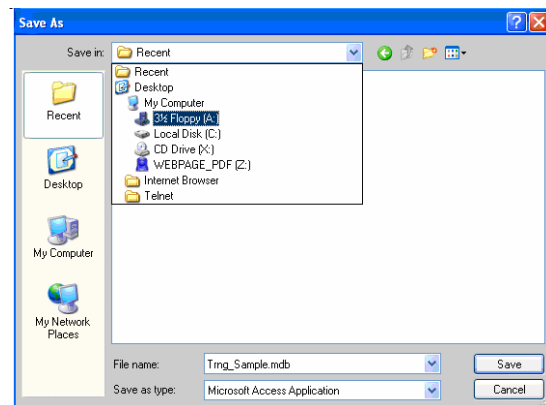


Figure 2 - Save As Dialog Box

4. Select a destination from the *Save in:* drop-down list box.
5. Use the default settings in the *File name:* and *Save as type:* text boxes ► **Save** button.

Creating a Macro

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 Design** window (see Figure 3). This window has two sections. The upper section, called the *Action* section, contains a design grid. The design grid can contain up to four 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* and *Condition* 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.

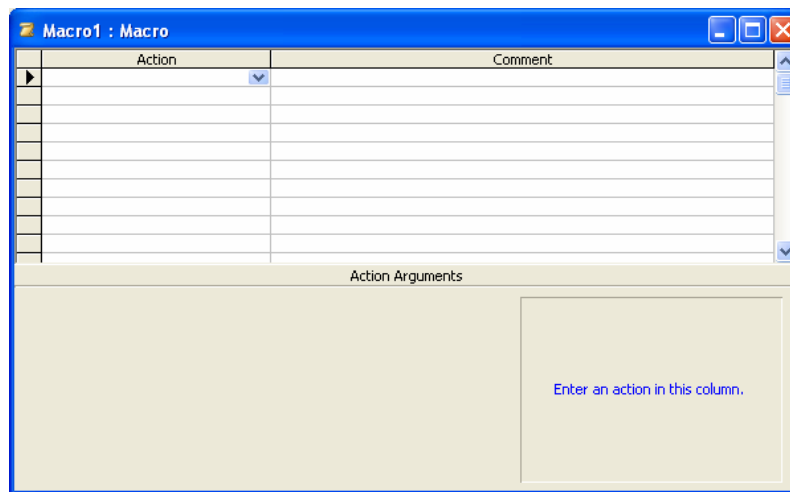


Figure 3 - The Macro Design Window

CREATING 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.

Other actions can be performed only in macros or in more complex programming modules. The “*AddMenu*” action, for example, creates a custom menu to appear on a custom menu bar. This action is not available in the menu commands.

A macro can include up to 999 actions. Each action is placed in a separate row in the design grid in sequential order. For example, a “*Maximize*” command maximizes the window opened in the step immediately preceding it.

The *Comment* column can contain up to 255 characters per line. 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 later.

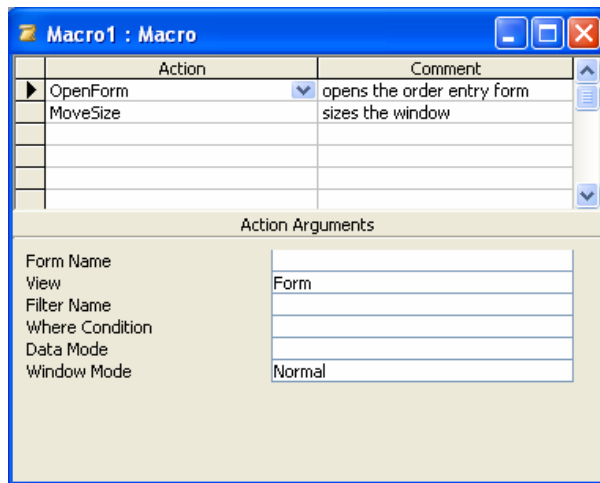





Figure 4 - Adding an Action and a Comment to a Macro

To create a macro:

1. Open the file named “**WORLD34.MDB**”.
2. Click the **Macros** button  to display the macros object in the database.
3. Click the **New** button . The **Macro Design** window will open.
4. Click in the first blank field in the **Action** column, if necessary.
5. Click the **Action** drop-down arrow .
6. Scroll as necessary and select the “**OpenForm**” command.
7. Click the corresponding field in the **Comment** column next to the “**OpenForm**” action.
8. Type [**Opens the Order Entry Form**] (Figure 4).
9. Add the “**MoveSize**” action to the second row of the **Action** column.
10. Enter the text [**Sizes the window**] in the corresponding field in the **Comment** column.

!NOTE:

Users can use the [Tab] key and the [Shift+Tab] key combination to navigate between the **Action** and **Comment** columns.


ASSIGNING AN ARGUMENT 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**,” and “**Print Preview**” arguments. For action arguments without a list, the argument can be 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 “[**Credit Limit**] = 1000” limits records

to those with a credit limit of \$1000. The condition can be typed directly into the argument text box or by clicking the **Build** button  to open the *Expression Builder* dialog box.

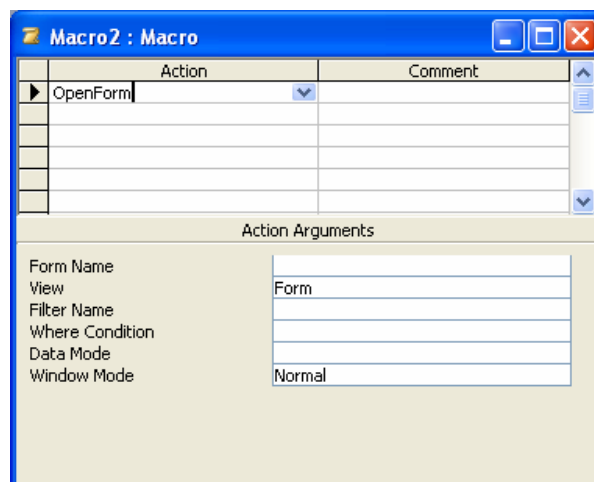



Figure 5 - Assigning an Argument to an Action

Access displays a helpful message explaining the selected argument in the right portion of the *Action Arguments* section. If the [F1] key is pressed while the insertion point is in the *Action Arguments* section of the **Macro Design** window, a help window for the argument opens with additional information.

To assign an argument to an action:

1. If necessary, open the **Macro Design** window.
2. Select the **“OpenForm”** action.
3. Click in the **“Form Name”** argument field in the *Action Arguments* section (Figure 5).
4. Click the **Form Name** drop-down arrow .
5. Select the **“Order Entry”** argument.
6. Set the argument **View** to **“Form,”** the argument **Data Mode** to **“Read Only”** and the argument **Window Mode** to **“Normal”**.
7. Click the **“MoveSize”** action in the *Action* section.
8. Set the arguments for **Right** to **“0”**, **Down** to **“0”**, **Width** to **“6.5”**, and **Height** to **“2.25.”**

!NOTE:


The [F6] key can be pressed to toggle between the *Action* section and the *Action Arguments* section. 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 save a macro or close the **Macro Design** window without saving, the *Save As* dialog box opens with the default name of **“Macro#”** (numbered consecutively). It is best to use a short name that clearly indicates the function of the macro. (The name **“Macro1,”** for example, does not mean much to other users who may be working with the macro).

The name that is given to the macro appears in the **Macros** object list in the **Database** window. All the macros associated with the database being used appear in the **Database** window.


To save a macro:

1. Click the **Save** button .
2. Type [**Open Order Form**] in the *Save As* dialog box ► **OK** button.

RUNNING A MACRO

A macro can be run from the **Database** window by selecting the **Macros** object button and double-clicking the desired macro name or by selecting the desired macro name and clicking the **Run** button  on the **Database** window toolbar.

To run a macro from the Database window:

1. If necessary, display the **Macros** object list in the **Database** window.
2. Click “**Open Order Form**” ► **Run** button .
3. After viewing the form, close the form window.


!NOTE:

A macro can be run by right-clicking the macro name in the **Database** window and selecting the **Run** command.

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. In the **Macro Design** window, the user can cut, copy, and paste to edit a macro. Rows can also be inserted and deleted. After the desired changes are made, the macro must be saved to retain the changes. 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. Select the “**Open Order Form**” macro.
2. Click the **Design** button .
3. Select the **OpenQuery** action in the *Actions* section under **MoveSize**.
4. In the comments pane type [**Open the Order Items query**].
5. In the *Action Arguments* section, set the “**Query Name**” argument to “**Order Items**” ► **Save** button.



CREATING A MACRO USING THE MACRO BUILDER

A macro can be created using the **Macro Builder**. The **Macro Builder** is used when the user wants to associate a macro with an existing object or control. The steps for creating a macro using this method are similar to creating a macro using the **New** button on the **Database** window toolbar.

Creating a macro using this method requires that the user associates the macro with an event, such as “**On Click**” or “**On Close**.” When an **Event** property is selected, the **Build** button appears. The **Build** button provides access to the **Macro Builder**.

To create a macro using the **Macro Builder**:

1. Click the **Forms** object button.
2. Open the “**New Entry**” form in **Design** view ► **Enter New Record** button.
3. Click the **Properties** button  ► **Event** tab.

4. Click the “**On Click**” property ► **Ellipsis** button .
5. Select “**Macro Builder**” ► **OK** button.
6. Type [**New Entry**] in the *Save As* dialog box ► **OK** button.
7. From the **Action** list, select “**GoToRecord**” ► type [**New Entry**] in the **Comment** column.
8. In the **Action Arguments** section, set the arguments for **Object Type** to “**Form**”, **Object Name** to “**New Entry**”, and **Record** to “**New**”.
9. Save the macro and close the *Macro Design* window.
10. Click the **Macros** object button and notice that a macro “**New Entry**” appears in the *Database* window.
11. Open the “**New Entry**” form in **Form** view (by clicking the **Open** button on the **Database** window toolbar). Observe the actions of the macro when executed by clicking the **Enter New Record** button .
12. Close the “**New Entry**” form ► close the “**WORLD34.MDB**” file.

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 properties in a single list.

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 users to control when an action occurs. The *Other* properties contain items that do not fit into the other three categories, such as the name of a control when used in a macro or text that appears in the **Status** bar.

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 utilize 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 “**WORLD35.MDB**”
2. Open the “**Customers**” form in **Design** view.

3. Click the “*Credit Limit*” text box ► **Properties** button  ► *Event* tab.
4. Click the *On Got Focus* property.
5. Click the *On Got Focus* drop-down arrow  ► select “*Limit*” ► **Close** button .
6. Close the “*Customers*” form and save the changes.
7. Open the “*Customers*” form again in **Form** view and click the “*Credit Limit*” field.
Notice that a message box generated by an existing macro opens.
8. Click the **OK** button to close the message box ► close the form.

ADDING A CONDITION TO A MACRO

A condition argument can be added to a macro. An argument acts very much like a filter. Just as a filter displays only the records that meet the condition, the argument only executes 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 Design** window. The *Condition* column appears to the left of the *Action* column in the upper pane of the **Macro Design** window. 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 with an exclamation point. For example, [Orders]![Customer ID] refers to the <i>Customer ID</i> field in the <i>Orders</i> table.
Controls in queries	Enclose the name of the query and the name of the control in square brackets and separate with an exclamation point. For example, [Order Items]![Item Number] refers to the <i>Item Number</i> field in the <i>Order Items</i> query.
Controls in forms	Enclose the name of the form and the name of the control in square brackets and separate with an exclamation point. Indicate that the reference is to a form by beginning the statement with the word “ Forms. ” For example, Forms![Customers]![Customer ID] refers to the <i>Customer ID</i> field in the <i>Customers</i> form.
Controls in reports	Enclose the name of the report and the name of the control in square brackets and separate with an exclamation point. Indicate that the reference is to a report by beginning the statement with the word “ Reports. ” For example, Reports![Customer Sales]![Contact Name] refers to the <i>Contact Name</i> field in the <i>Customer Sales</i> 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.

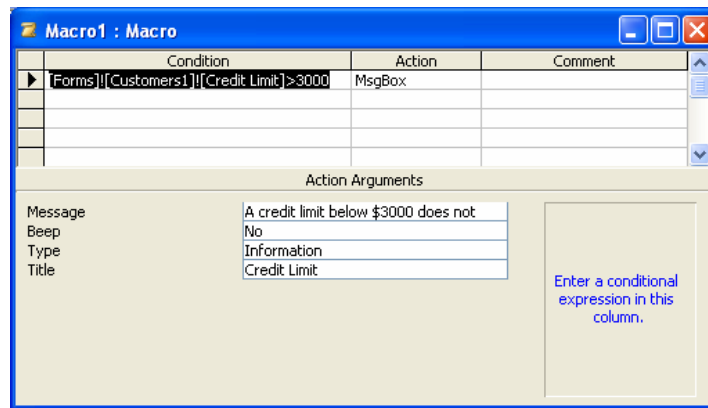



Figure 6 - Adding a Condition to a Macro

!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. Click the **Conditions** button  on the **Macro Design** toolbar.
3. Click in the **Condition** column next to the “**MsgBox**” action.
4. Type “[**Forms**]![**Customers1**]![**Credit Limit**]<3000” (include the brackets) (Figure 6). Expand the **Condition** column, if necessary.
5. Close the **Macro Design** window and save the macro.
6. 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 above \$3000.
7. Move to record “5”. Notice that the message box opens because the credit limit for this particular customer is under “\$3000.”
8. Click the **OK** button to close the message box ► close the form.

Using Other Form Techniques

CREATING A PIVOTTABLE OR PIVOTCHART VIEW

Any database form can be viewed in **PivotTable View** or **PivotChart View** using the **View list**  when the form is open. In **PivotTable View**, large amounts of data can be summarized and analyzed. **PivotChart View** allows the user to display data in a graphic environment. 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 bold 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 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.

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

1. Close “**WORLD35.MDB**” ► Open “**WORLD29.MDB**.”
2. Open the “**Customer Sales Information**” form in **Design** view.
3. Click the **View** button drop-down arrow ► “**PivotTable View**.”
4. Click the **Field List** button on the **Form Design** toolbar if necessary ►
 Store Name
5. Drag the “**Store Name**” field to the “**Drop Row Fields Here**” area ►  Customer Type
6. Drag the “**Customer Type**” field to the “**Drop Column Fields Here**” area ►
 Credit Limit



Store Name	Customer Type		
	1	2	3
Ace Sporting Goods			5000
AI's Sporting Goods	1200		
Alvarez Equipaje de Juegos		2000	
Athlete's Dream			3500
Athlete's World	1500		
Athletic Supplies Co.			8000
B&B Sporting Goods	2300		
Big Marty's Sports		7000	
Canadian Sports Ltd.	3000		
Champion Sports Equipment			8500
Champs			2300
Classic Sports Co.	1500		
Coastal Athletic Supply House			

Figure 7 - Data in Pivot Table View

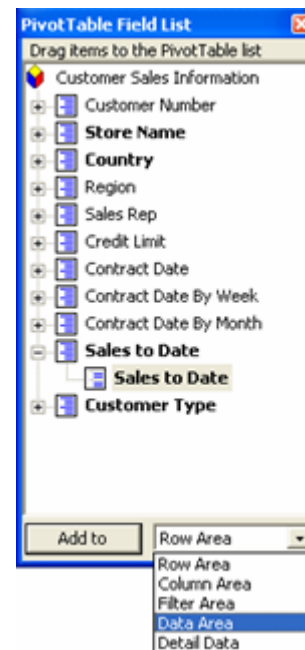



Figure 8 - Pivot Table Field List

7. Drag the “**Credit Limit**” field to the “**Drop Totals or Detail Fields Here**” area ►
 Country
8. Drag the “**Country**” field to the “**Drop Filter Fields Here**” area
9. Switch to **PivotChart View**. Notice that the data area is blank. That is because the “**Credit Limit**” field was added by dragging and is displayed as a detail data type.
10. Switch back to **PivotTable View**.

11. Delete the “**Credit Limit**” field by right-clicking on the title “**Credit Limit**” and selecting **Remove** from the short-cut menu ► Select the “**Sales to Date**” field in the **Field List**.
12. Select “**Data Area**” from the drop-down list box at the bottom of the **Field List**. (see Figure 8) ► Click the **Add to** button to add the field to the view workspace.
13. Switch to **PivotChart View** and notice that the sales to date data are displayed for each store (see Figure 9) ► Switch to **PivotTable View**.
14. Select the “**Store Name**” row heading ► Bold and italicize the store names.
15. Switch back to **PivotChart View** and notice that the store name formatting does not display. ► Close the form and do not save the changes.
16. Close “**WORLD29.MDB**” ► Open “**Trng_Sample-Adv.mdb.**”

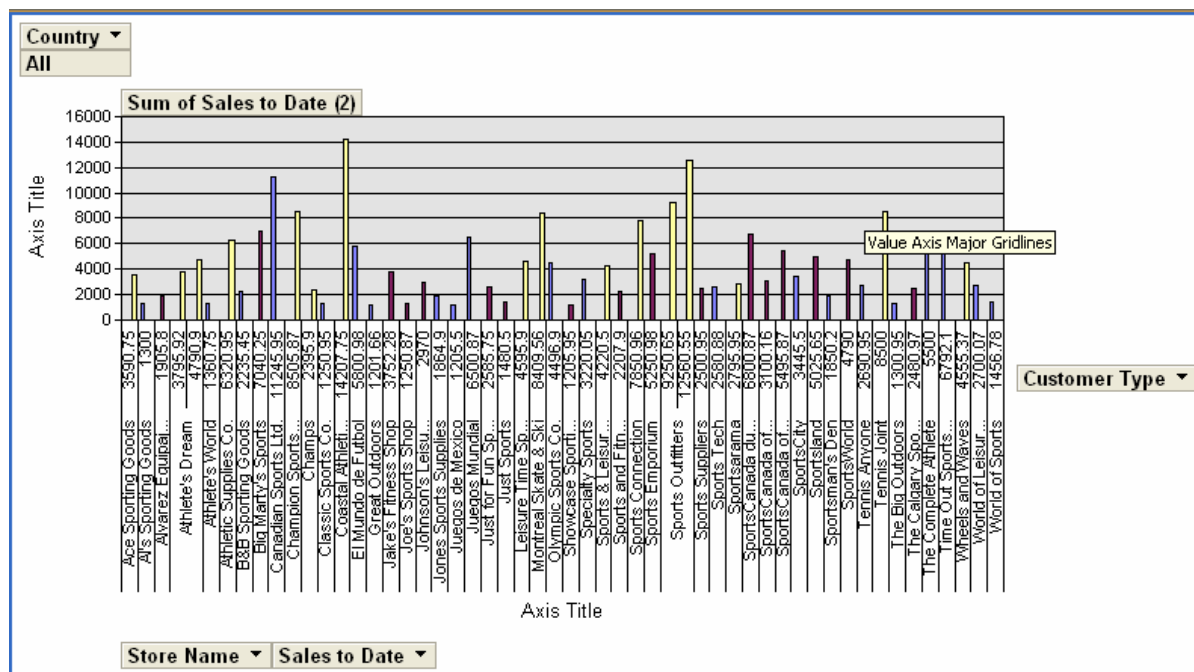


Figure 9 - Data in Pivot Chart View


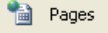
Using Data Access Pages

A **Data Access Page** is a special type of Web page which can be used to add, edit, view, or manipulate data in an existing Access database. Each **Data Access Page** is saved as a separate file, which is connected to an Access database. Therefore, if one user views the **Data Access Page** in a Web browser and makes changes (sorting or filtering), it only affects that particular page, and not the database file itself or other users. However, if changes are made (adding, deleting, editing records) in a database file, it affects the database file as well as the files of other users. The database file can be transferred to a Web server via the Internet or an intranet.

CREATING A DATA ACCESS PAGE

There are several different ways to create a data access page. One of the simplest is to use the **Page Wizard**. The **Page Wizard** asks detailed questions about the desired record sources, fields, layouts, and format, then creates a **Data Access Page** based on the answers.

To create a new page using the Page Wizard:

1. Go to the **Database** window
2. Select the **New** button  from the **Pages** object list . The **New Data Access Page** dialog box will appear.

3. Select **Page Wizard** ► **OK** button (see Figure 10).

!NOTE:

The **Page Wizard** begins and the user follows the steps to create a new page. The user should choose the table or query on which the **Data Access Page** is to be based as well as particular field names that are to be added in the page. The user may also select several fields together from more than one table (or query) that are related to each other.

4. Select "**Table: StudentRecord**" in the **Tables/Queries** drop-down list box.
5. Add the desired fields to the **Selected Field** list (see Figure 11) ► **Next >** button.

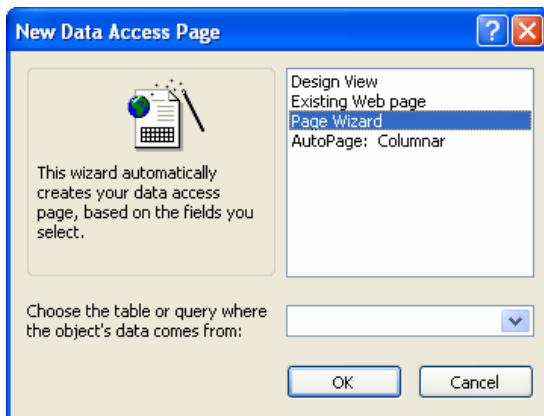


Figure 10 - New Data Access Dialog Box

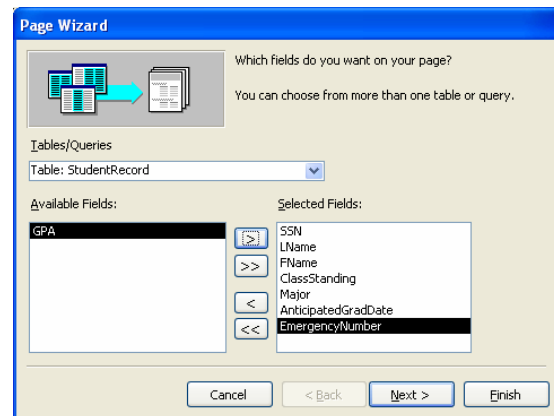


Figure 11 - Page Wizard: Select Fields for a Page

6. Select the "**Major**" field as a grouping level (see Figure 12). In this way, users will get the contact information for each student grouped by major ► **Next >** button.
7. Select the **I** drop-down arrow.
8. Select the "**LName**" field as a sorting key and click the **Ascending** button if necessary (see Figure 13). All records will sort alphabetically in ascending order ► **Next >** button.

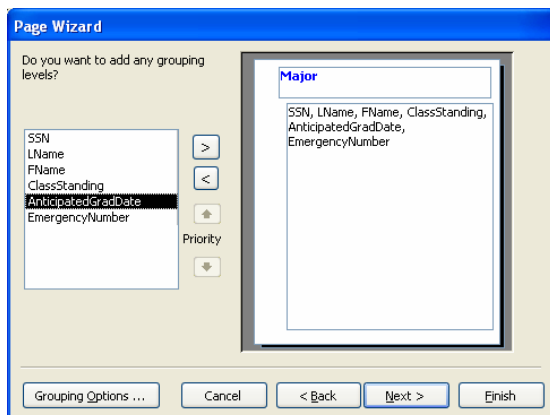


Figure 12 - Page Wizard: Select Grouping Field(s)

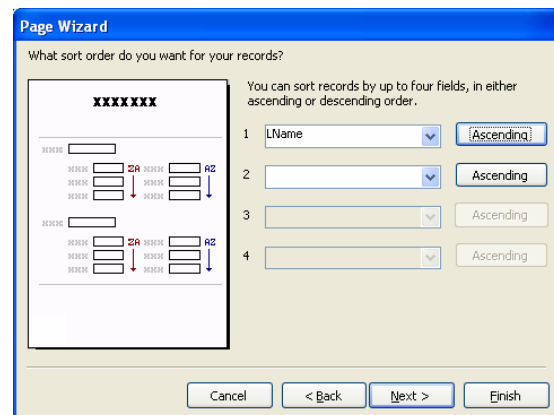


Figure 13 - Page Wizard: Sort Order for Records

9. Type **[StudentRecord db-page]** in the **What title do you want for your page?** text box (see Figure 14).
10. Click the **Finish** button. A new **Access Page** window appears (see Figure 15).

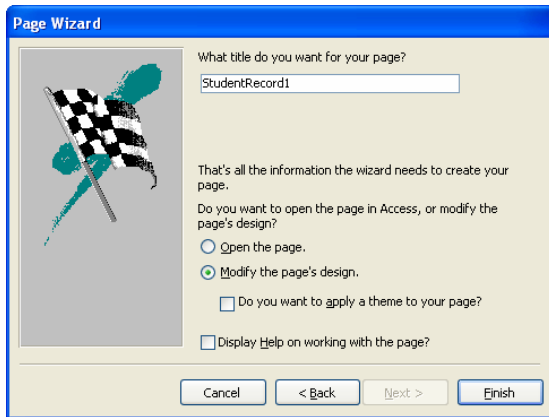


Figure 14 - Page Wizard: Save a Name of New Page

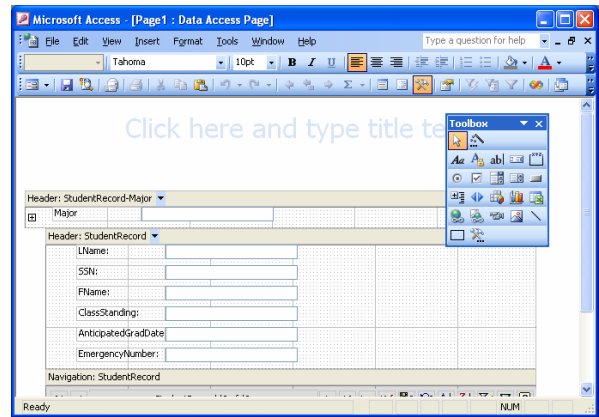


Figure 15 - New Page in Design View

!NOTE:

The new page displays the records in **Design View**. To view and operate in **Page View**, click the **Toolbar** button or menu (Page View). Figure 16 shows the interface of **Data Access Page** and descriptions of each component.

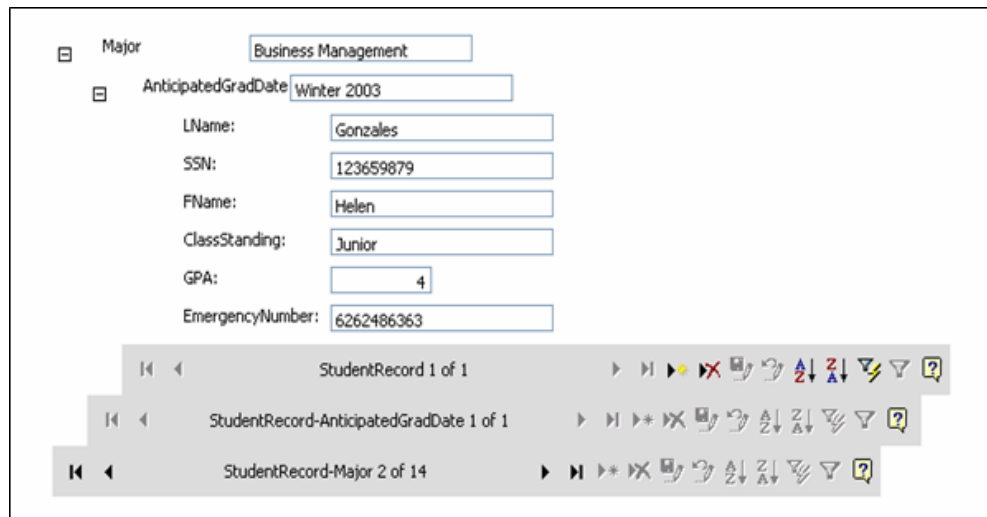


Figure 16 - Access Page View

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/SUBREPORT

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 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 Wizard* and *Subreport Wizard* provide the easiest method for creating a *Subform* or *Subreport*. If the user chooses not to use the wizards, a form/report will need to be designed and then inserted into another form/report. The wizard need not be used if a *Subform/Subreport* being designed is very different in appearance from what the wizard would produce. However, the wizards not only makes it easy 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 the “*Contacts*” and “*Enrollments*” tables are related with a common field, “*SSN*”. The “*Contacts_Report*” is created from the “*Contacts*” table. A *Subreport* may be created in “*Contacts_Report*” to display individual enrollment records within one *Report View*. As shown below, the relationship between the “*Contact*” and “*Enrollments*” tables displayed in table *Datasheet View* (Figure 17) is also printable in *Report Print Preview* (Figure 18).

Contacts : Table				
	SSN	LName	FName	Tit
▶	100-25-9487	Kim	David	Second
▶	EnrollmentID	EnrollmentDate	ClassID	
▶	3544	Fall 2002	11254	
▶	6587	Fall 2002	13795	
▶	1973	Fall 2002	14789	
▶	1268	Fall 2002	15364	
*				

Figure 17 - Relationships in Table Datasheet View




SSN 100-25-9487						
LName	FName	Title	Address	City	PostalCode	Home
Kim	David	Second Trainer	1245 Common West	Alhambra	91803-2548	(626) 5
Enrollment subreport						
EnrollmentID	SSN	EnrollmentDate	ClassID			
1268	100-25-9487	Fall 2002	15364			
6587	100-25-9487	Fall 2002	13795			
1973	100-25-9487	Fall 2002	14789			
3544	100-25-9487	Fall 2002	11254			

Figure 18 - Report Print Preview with SubReport

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*, a floating toolbox opens on the screen. If not, select the **View** menu ► **Toolbox** (or click the **Toolbox** button  on the toolbar).
 - Check if the **Control Wizards** button  is activated on the toolbox.
2. Click the **Subform/Subreport** button  on the **Toolbox** toolbar. Users may notice that the mouse pointer has changed into the **Subform/Subreport** tool button with a plus sign.

- Place the pointer to insert a **Subreport** in the **Detail** section. The **SubReport Wizard** will open (see Figure 19).
- Click the **Use existing Tables and Queries** option button to select original data on which to base the **Subreport** ► **Next** > button.
- Select the “**Enrollment**” table from the **Tables/Queries** drop-down list.
- Transfer all fields to the selected fields using double rightward-pointing arrow (see Figure 20) ► **Next** > button.

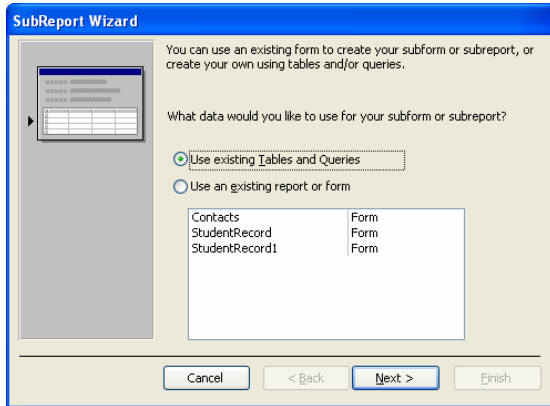


Figure 19 - SubReport Wizard: Step 1

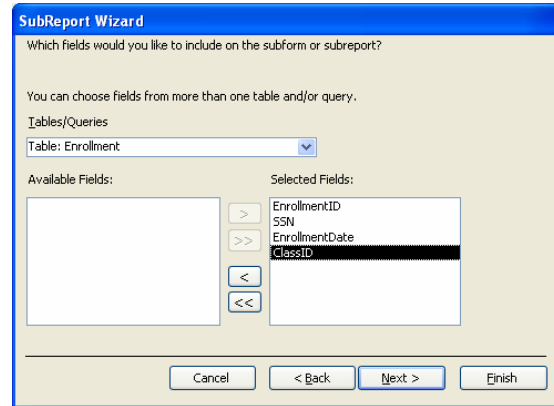


Figure 20 - SubReport Wizard: Step 2

- 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.
- Name the **Subreport** ► **Finish** button (see Figure 22).

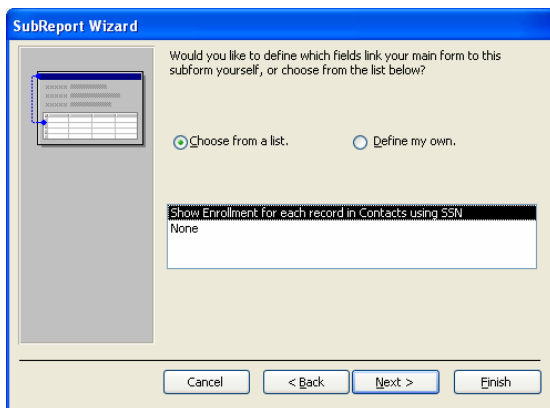


Figure 21 - SubReport Wizard: Step 3

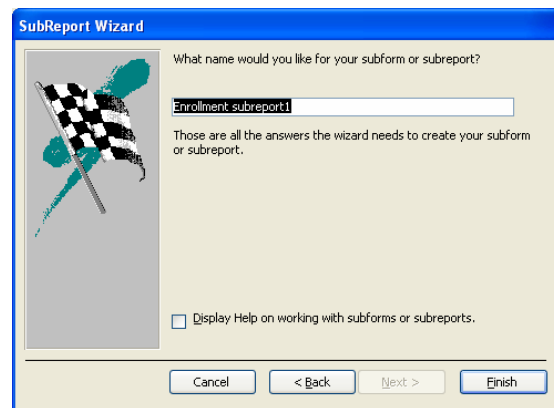


Figure 22 - SubReport Wizard: Step 4

Figure 23 shows “**Contacts_Report**” including the “**Enrollment Subreport**” in **Design View**. If it is viewed in **Print Preview**, it should be the same as Figure 18 above.

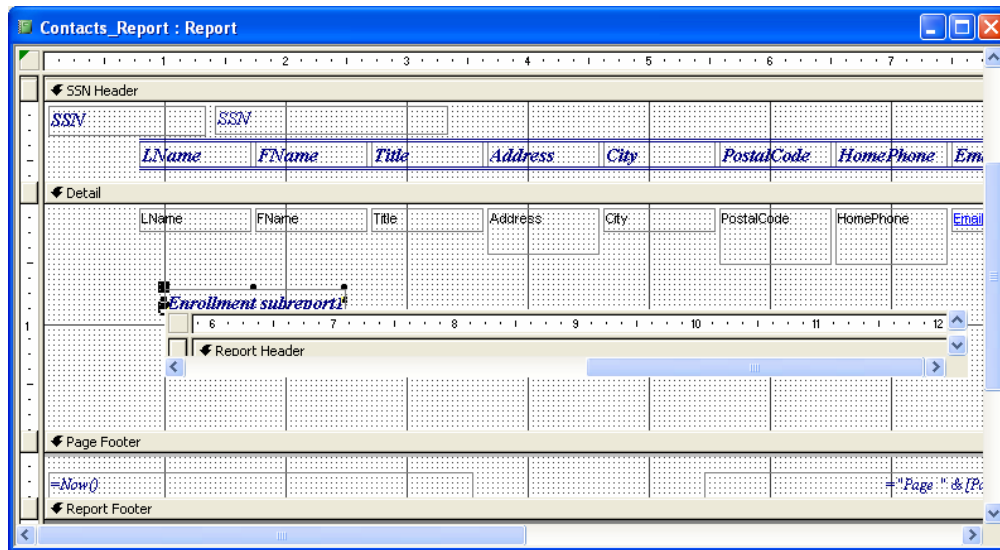


Figure 23 - Subreport within a Report in Design View

CHANGING LAYOUT OF A SUBFORM

To improve the efficiency or appearance of a *Subform/Subreport*, the 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 like 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 24).

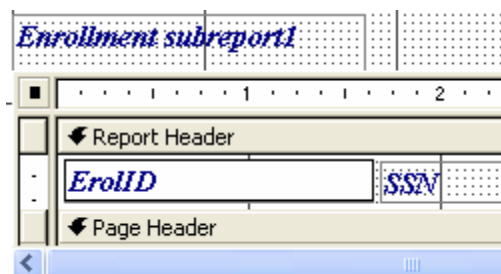


Figure 24 - Change layout of SubReport


Generating Multicolumn Reports as Mailing Labels

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

CREATING MAILING LABEL WITH LABEL WIZARD

Mailing lists can be created with the **Label Wizard** or 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 detail section of the report. The **Label Wizard** can also be customized for labels with unusual sizes or those produced by manufacturers that are not 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. With the **Report** object selected click the **New** button  on the toolbar. The **New Report** dialog box opens (see Figure 25).
2. Select **Label Wizard** in the list then select the “**Contacts**” table (from the drop-down list box) as the data source for the labels ► **OK** button. The **Label Wizard** opens (see Figure 26).
3. Select “**Avery labels**” and the code “**5260.**” Click the **Sheet Feed** option button if laser-printer labels are being used ► **Next >** button.

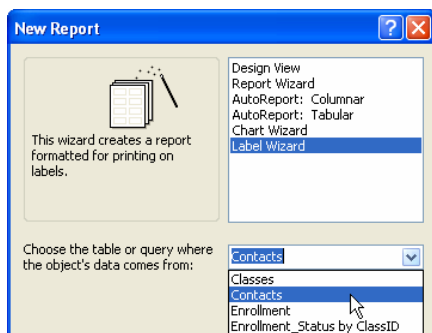


Figure 25 – New Report Dialog Box

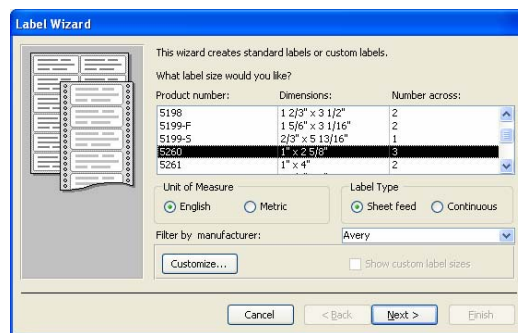


Figure 26 - Label Wizard: Step 1

4. Select “**Courier New**” in the **Font name** drop-down list box, “**Medium**” in the **Font weight** drop-down list box and “**9**” in the **Font size** drop-down list box (see Figure 27) ► **Next >** button.
5. For the first row of the label, select “**SSN**” from the **Available fields:** list box.
6. Move the field to the **Prototype Label:** text box by clicking the single, right-pointing arrow button ► press [Enter] to add a new line.
7. Select “**LName**” from the field list and add it to the prototype ► add a comma and a space.
8. Add “**FName**” to the prototype ► press [Enter] to add a new line.
9. Add “**HomePhone**” to the prototype ► **Next >** button (see Figure 28).

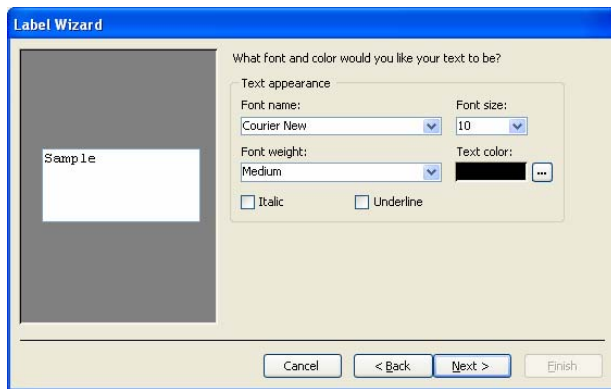


Figure 27 - Label Wizard: Step 2

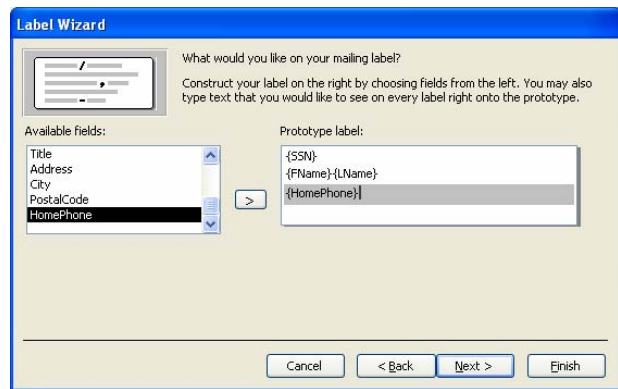


Figure 28 - Label Wizard: Step 3

10. Select "*SSN*" from the *Available fields*: list box and move it to the *Sort by*: list box ► **Next >** button (see Figure 29).
11. Name the report ► **Finish** button (see Figure 30).

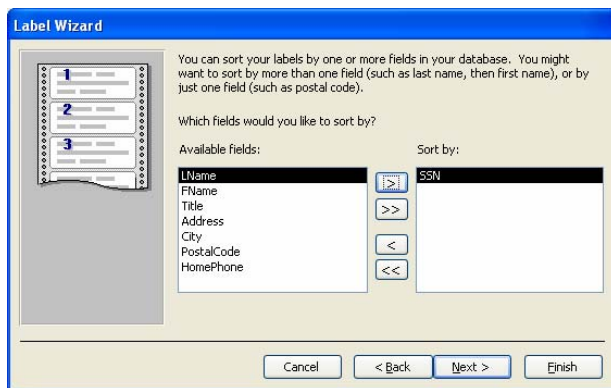


Figure 29 - Label Wizard: Step 4

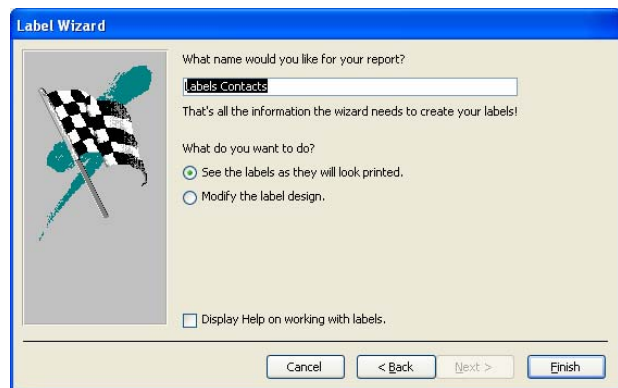


Figure 30 - Label Wizard: Step 5

!NOTE:

In many cases an error message as shown in Figure 31 appears before opening the labels in **Print Preview** (see Figure 32). This is due to the *Wizard* miscalculating the column widths. Click the **OK** button to dismiss the message and display the labels in **Print Preview**.

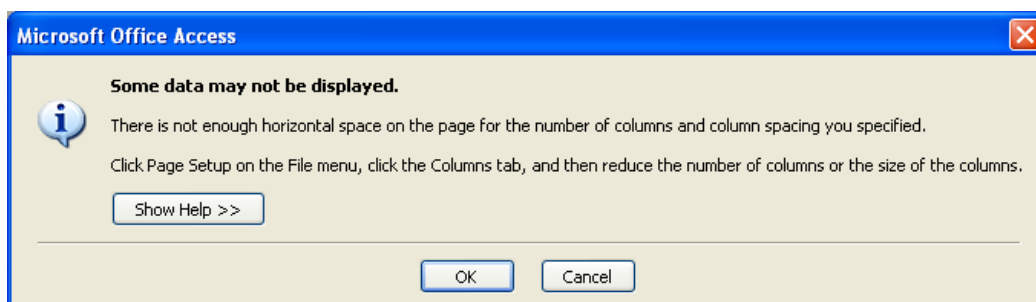


Figure 31 - Error Message when Creating a Label using Wizard

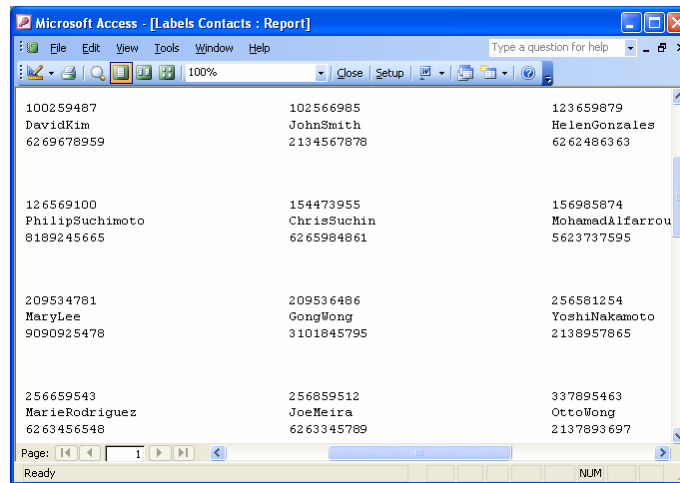


Figure 32 - Labels in Report Print Preview: Created by label Wizard

Creating Custom Toolbars

When creating a custom application, the user might also want to create custom menus. Creating custom menus allows the user to limit the actions to be performed. For example, certain commands can be removed from the **File** menu such as **New...** and **Save As...** to prevent users from modifying the application database. A custom command can also be added that initiates a macro.

CREATE A CUSTOM TOOLBAR

Menus and command buttons reside on toolbars. A toolbar usually contains only menus or only buttons although both can be displayed on the same toolbar. Once a custom toolbar has been created, it can be linked to a form or a report, meaning that it will automatically appear when the form or report is opened or run.

!NOTE:

The toolbar will automatically resize when custom menus and buttons are added.

To create a custom toolbar:

1. Select the **View** menu ► **Toolbars** ► **Customize...**. The *Customize* dialog box opens (see Figure 33).
2. Select the *Toolbars* tab ► **New...** button. The *New Toolbar* dialog box appears in which users can assign a name to the new toolbar (see Figure 34).
3. Name the new toolbar "**Student Report toolbar**" ► **OK** button ► **Close** button.

!NOTE:

Once a customized toolbar is created, it opens a floating toolbar named **Student Report** toolbar. Buttons can be added or removed on the custom toolbar designed by the user (see Figure 35).

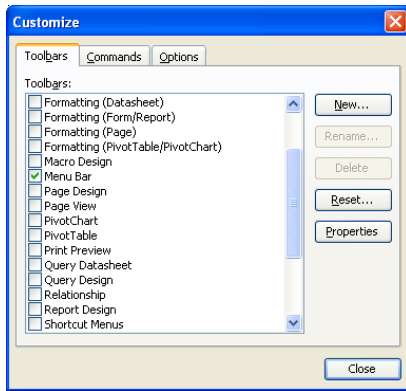


Figure 33 - Customize Dialog Box

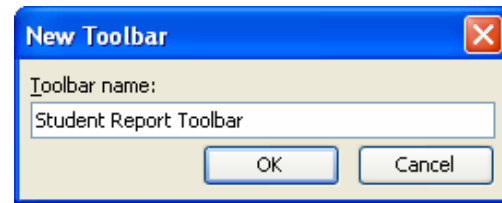


Figure 34 - New Toolbar Dialog Box

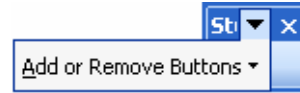


Figure 35 - Customize Toolbar

ADDING A BUILT-IN MENU ITEM

As stated above, any toolbar can be displayed and edited through the *Customize* dialog box. Menu items are listed by category, each of which contains a group of related menu commands. For example, the “*Edit*” category, when selected, displays editing commands such as “*Copy*” and “*Paste*.” Entire menu items or individual menu commands can be quickly added or removed to and from existing toolbars in Access. For example, the *File* menu can be removed to prevent users from performing any file operations. The *Customize* dialog box includes built-in menus that can be added to a toolbar. These built-in menu items already include a group of commands.

To add a built-in menu item to the customized toolbar:

1. Select the **V**iew menu ► **T**oolbars ► **C**ustomize.... The *Customize* dialog box opens.
2. Select the *Toolbars* tab ► click the **S**tudent Report Toolbar check box.
3. Select the *Commands* tab ► click **B**uilt-in Menus in the *Categories* list box.
4. Click **V**iew in the *Commands* list box and drag it to the **S**tudent Report Toolbar. The **S**tudent Report Toolbar now contains the **V**iew menu as shown in Figure 36 ► **C**lose button.

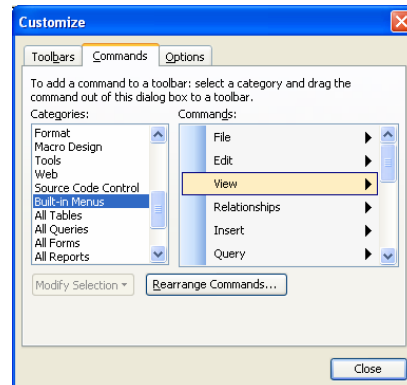


Figure 36 - Customize Dialog Box: Adding Built-in Menu



CREATING A CUSTOM MENU ITEM

Adding built-in menus is a quick way to develop a custom toolbar. Built-in menus are menu items that already contain a group of commands. However, the user can create a personal menu item to which a group of commands can be assigned. Access allows users to create a new menu item to which a name can be assigned and commands can be added that appear in a list when the menu is selected.

To create a custom menu item:

1. Open the *Customize* dialog box and display the **Student Report** Toolbar (see Figure 37).
2. Select the *Commands* tab ► click **New Menu** in the *Categories:* list box.
3. Click **New Menu** in the *Commands* list box.
4. Drag the **New Menu** menu to the right of the **View** menu on the **Student Report** Toolbar ► **Modify Selection** button.
5. Drag to select the text “**New Menu**” in the *Name* text box ► type [**Enrollment**] (see Figure 38) ► press the [**Enter**] key ► **Close** button.

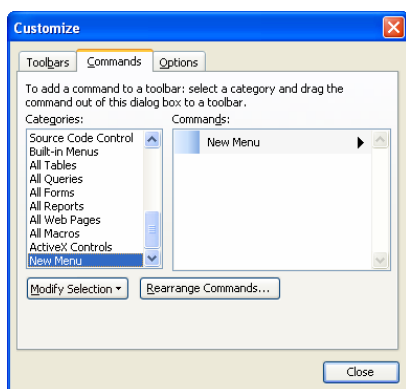


Figure 37 - Customize Dialog Box: Adding Menu Item

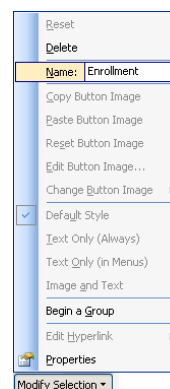


Figure 38 - Modifying Selection

ADDING A COMMAND TO A MENU ITEM

The *Customize* dialog box contains commonly used commands that may also be added to a menu item if they are not already there. When adding a command, a vertical or horizontal bar appears, indicating where the command will be added.

To add a command to a menu item:

- Open the *Customize* dialog box and display the **Student Report** Toolbar (see Figure 39).

!NOTE:

The custom toolbar is created as a floating toolbar by default, meaning that it appears as a small window that can be easily moved and closed. Toolbars can be docked horizontally along the top or bottom of the Access window, or vertically along either side of the window. Users can place the custom **Student Report** Toolbar in any preferred location.

1. Select the *Commands* tab ► click **File** in the *Categories:* list box.
2. Click **Page Setup...** in the *Commands:* list box and drag it to the **Enrollment** menu until the **I-beam** appears in the desired location (see Figure 39).
3. Add the **Print Preview** and **Print** commands to the **Enrollment** toolbar.

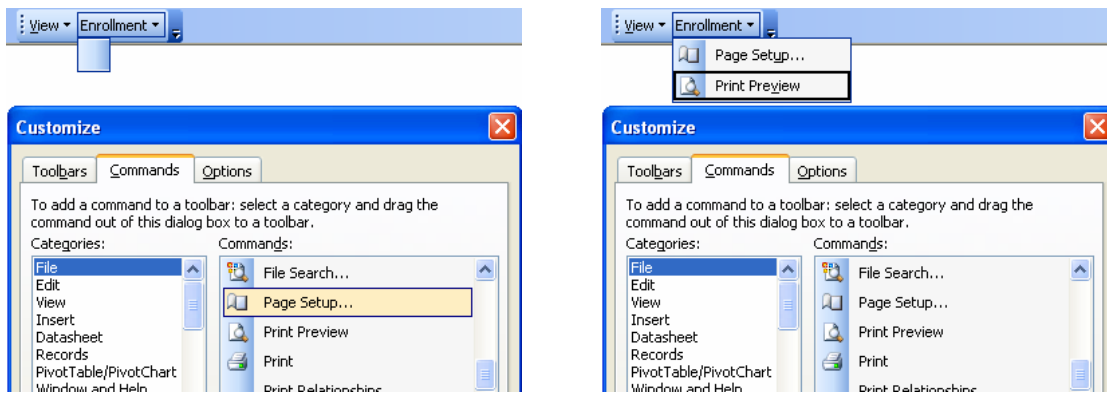


Figure 39 - Adding Commands to a Menu Item

ADDING CUSTOM COMMANDS

Most of the commands available in the *Customize* dialog box have an associated action. For example, the **Print** command performs the action of printing the active object, such as a form or report. However, the **Custom** command, which is included in the selection of available commands, has no action associated with it when added to a menu item. A macro can be assigned to this command so that when it is selected, it performs the actions of that macro. The **Custom** command can also be renamed to provide a description of the actions that the command performs.

To add a custom command:

1. Open the *Customize* dialog box and display the **Student Report** Toolbar.
2. Select the *Commands* tab ► click **File** in the **Categories:** list box ► click **Custom** in the **Commands:** list box.
3. Drag the **Custom** command to the **Enrollment** menu until the **I-beam** appears at the bottom of the menu (see Figure 40).
4. Click the **Modify Selection** button ► select **Custom** in the **Name:** text box.
5. Type **[Print Contact_Report]** (see Figure 40) ► click **Properties**. The *Custom Popup* dialog box opens (see Figure 41).
6. Click the **On Action:** drop-down list box ► select **Print_Contacts_Report** ► **Close** button.

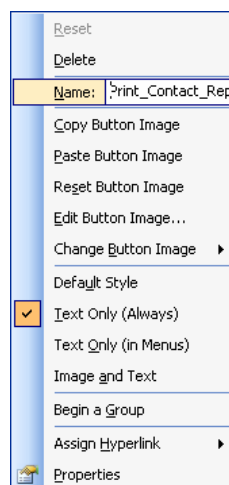


Figure 40 - Adding Custom Command

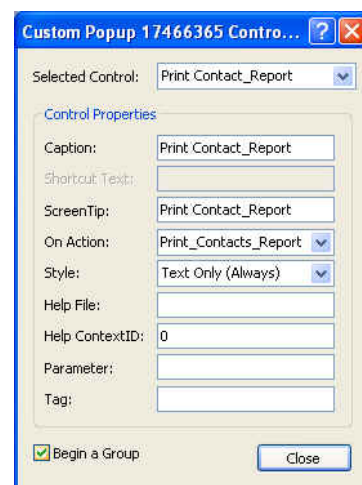


Figure 41 - Custom Popup Dialog Box


!NOTE:

Through the steps above, a **Custom** menu command named **Print Contact_Report** is created. Whenever the command is clicked it runs the macro associated with it.

LINKING A TOOLBAR TO A REPORT

After a custom toolbar has been created, it can be linked to a form or report. When the form or report opens, the custom toolbar appears along with any default toolbars.

To link a toolbar to a report:

1. Display the **Reports** object list.
2. Open the “**Enrollment Subreport**” in **Design View**.
3. Click the **Properties**  button on the **Report Design** Toolbar. The *Properties* dialog box opens (see Figure 42).
4. Select the *Other* tab ► select the **Student Report** Toolbar by clicking the **Toolbar** drop-down list box ► close the *Properties* dialog box and save the changes.

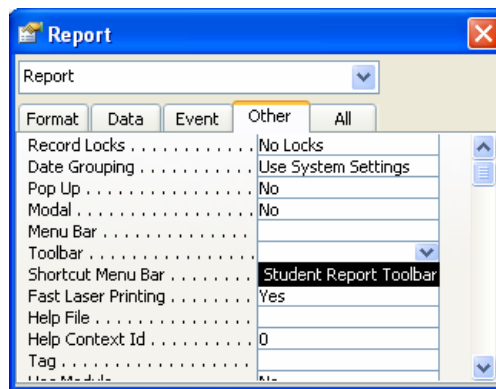


Figure 42 - Report Properties Dialog Box

ADDING A SEPARATOR BAR

A separator bar is added on menus to group related commands. For example, the **File** menu in Access has several separator bars that distinguish groups of file operations, such as opening, closing, saving, and printing. Separator bars can be added to menus the user has created to group related commands.

To add a separator bar in a custom toolbar:

1. Display the *Customize* dialog box and display the **Student Report** Toolbar.
2. Click **Enrollment** ► **Print Contact_Report** ► **Modify Selection** button in the *Commands:* tab of the *Customize* dialog box.
3. Click **Begin a Group**. Users may see the separator between two commands (**Print Preview** and **Print Contact_Report**). This means several commands are grouped (see Figure 43).
4. Close the *Customize* dialog box.

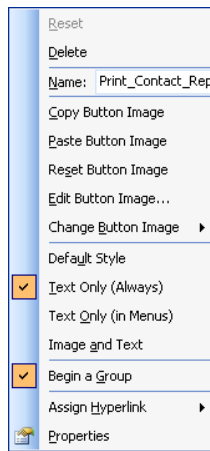


Figure 43 - Adding a Separator Bar

ADDING BUTTONS TO A CUSTOM TOOLBAR

Command buttons can be added to a custom toolbar. When clicked, command buttons perform an action such as saving a file or opening a database object. Generally the action taken is depicted in a graphic on the button. For example, the **Save** button has a graphic of a floppy disk on it. The *Customize* dialog box allows the user to select command buttons from several categories. A toolbar can be created that contains only command buttons, or menu items can be mixed with command buttons. The items listed in the **Commands:** list box of the *Customize* dialog box displays the text for the commands as well as the corresponding buttons.

To add buttons to a custom toolbar:

1. Display the *Customize* dialog box.
2. Select the **Commands:** tab ► **File** (in the **Categories:** list box).
3. Click **Save** in the **Commands:** list box and drag it to the right of the **Enrollment** menu on the **Student Report** Toolbar (see Figure 44).

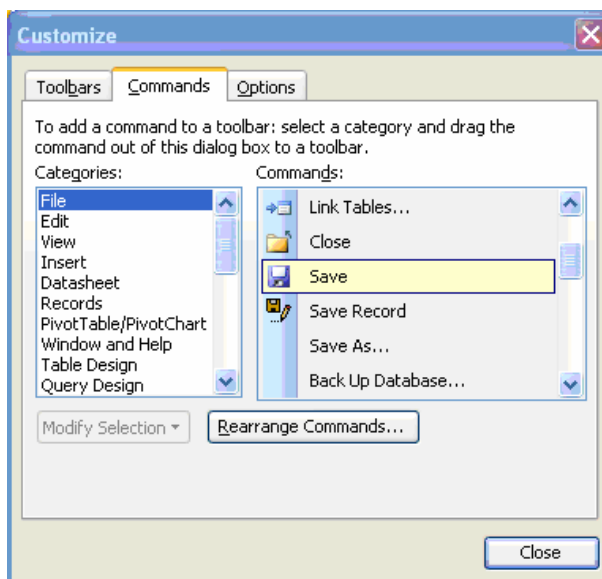


Figure 44 - Adding buttons in a custom toolbar

DELETING A CUSTOM TOOLBAR

A custom toolbar can be deleted when it is no longer needed.

To delete a custom toolbar:

1. Display the *Customize* dialog box.
2. Select the *Toolbars* tab ► *Student Report Toolbar* ► **Delete** button ► **OK** button ► **Close** button.

Using Advanced Database Features

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.

A copy of the original database can be created by saving the compacted database with a different name or to a different location. Although the current database can be compacted while it is open, the database must be closed to make a copy.

To compact a database:

- Select the **T**ools menu ► **D**atabase Utilities ► **C**ompact and Repair Database....

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 the 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.

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 45 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 2003 databases. However, if the user is working in a converted database, it must be manually enabled.

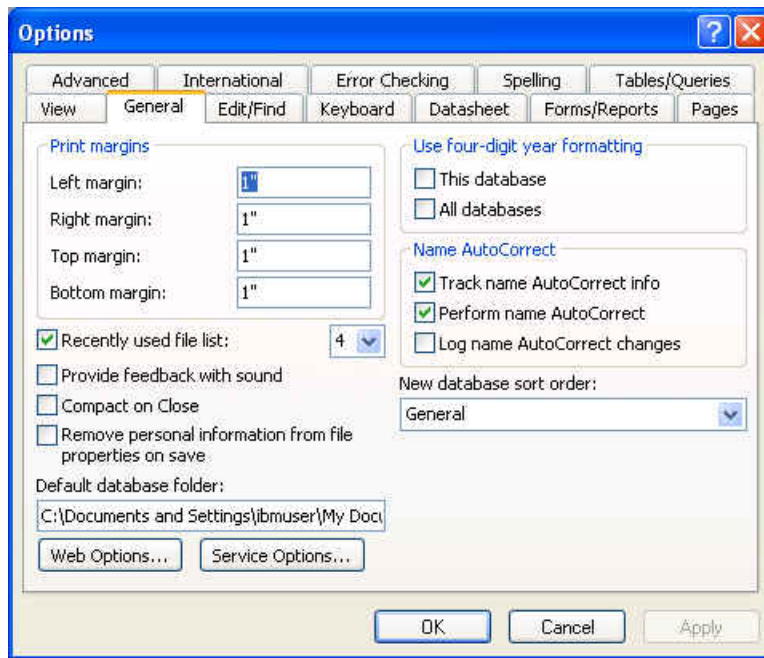


Figure 45 - Options dialog box: Name AutoCorrect

To deactivate the *Name AutoCorrect*:

1. Select the **T**ools menu ► **O**ptions.... The *Options* dialog box opens (see Figure 45).
2. Select the *General* tab.
3. Deselect the *Track name AutoCorrect info* and *Perform name AutoCorrect* check boxes
► **A**pply button ► **O**K button.

!NOTE:

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 *FN*ame field to *First*Name.
- 3) Save the changes and close the table.
- 4) Open the “*Contacts*” form in *Form View*.

Notice that Access is unable to find values for the *FN*ame field from original “*Contacts*” table information. 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. Open the *Options* dialog ► select the *General* tab.
2. Click the *Track name AutoCorrect info* and *Perform name AutoCorrect* check boxes.
3. Open the “*Contacts*” table in *Datasheet View*.
4. Change the *Address* field to *Add*.
5. Save the changes and close the table.
6. Open the “*Contacts*” form in *Form View*.

Notice that this time Access is able to find values for the renamed field.

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. Such information can be viewed and printed as the properties associated with tables, queries, forms, and reports. This type of information is very useful for deciding what changes are to be made or when maintaining a database developed by someone else.

To preview the structure of relationship from current database file:

1. Select the **Tools** menu ► **Aalyze** ► **Documenter**.
2. Select the *Current Database* tab ► ***Relationships*** check box ► **OK** button.

!NOTE:

Click the **Print Preview** button  from **Standard** toolbar before printing.

Appendix 1 - Commonly Used Actions and Descriptions in MS Access Macros

Action	Description
ApplyFilter	Applies a filter or query to restrict or sort records.
Beep	Sounds a tone through the computer speaker.
Close	Closes a specified window.
CopyObject	Copies the specified object to a different database or the same database under a new name.
DeleteObject	Deletes the specified object.
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.
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.
OpenDataAccessPage	Opens a Data Access Page in the selected view.
OpenForm	Opens a form in the selected view.
OpenModule	Opens a module at the selected procedure.
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).
PrintOut	Prints the specified datasheet, form, report, or module.
Quit	Quits Access.
Rename	Renames the specified database.
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.
RunApp	Runs an application from within Access.
RunMacro	Runs a macro.
RunSQL	Runs an action query using the corresponding SQL statement.
Save	Saves the specified object. If an object is not specified, the active object is saved.
SelectObject	Selects an object.
SetValue	Sets the value of a field, control, or property on a form or report.
ShowAllRecords	Removes any applied filter from the applicable active object and displays all records.
ShowToolbar	Displays or hides a toolbar.
StopAllMacros	Stops all currently running macros.
StopMacro	Stops the currently running macro.
TransferText	Imports or exports text between the active database and a text file.