

**OREGON STATE  
UNIVERSITY**

**ECTU**

**Extension Computing Technology Unit**

**Advanced  
Access 2000**

**Interfacing Word and Outlook**

by

**Kevin E. Presley  
Training Coordinator**

**September, 2000**



---

<b>Creating Basic Database</b>	<b>1</b>
<b>Creating Query Selection Form</b>	<b>2</b>
<b>Creating the Query</b>	<b>3</b>
<b>Moving on to Word</b>	<b>5</b>
<b>Mailing Labels</b>	<b>6</b>
<b>Form Letters</b>	<b>8</b>
<b>Creating Buttons on Query Form</b>	<b>9</b>
<b>Outlook interfaces</b>	<b>11</b>



## Creating a Basic Database

The database used in this material is just a simple mailing list database created by using the Wizards in Access. The only additions to this database are the additions of the Group category check boxes. The fields selected from the Wizard are: MailingListID, Prefix (Mr., Mrs., Miss, etc.), FirstName, MiddleName, LastName, Suffix (II, III, etc.), Title, OrganizationName, Address, Region (which was changed to County), City, State, PostalCode, HomePhone, WorkPhone, FaxNumber, EmailAddress, and Notes. The added fields are YESNO data type: 4H, Master Gardener, Master Food, OCA(Oregon Cattleman’s Association).

This database can be found at:

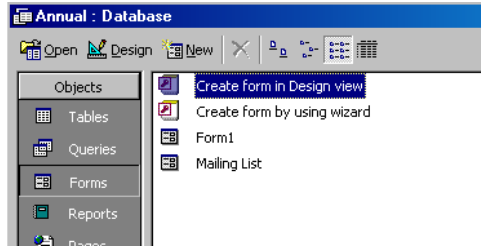
<http://osu.orst.edu/extension/ectu/training/lessons/Annual/Annual2000.zip>

This training will not cover creating that database or an input form for entering the data to the Mailing List table. If you need instruction on doing these steps, see the Intermediate training for Access. The form in the database used in this training for input was designed using the Wizard. Extract the files into a folder called Annual.

## Creating a Query Selection Form

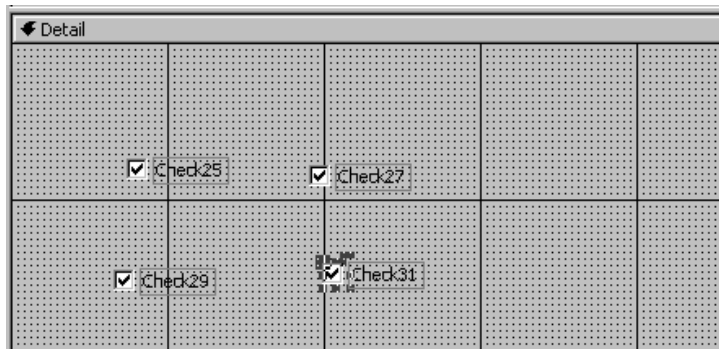
Go to the Forms tab to complete this process.

The title above is actually a combination of a two-step process. This form will enable us to use a query to select different listings for form letters and mailings. The first step is to create a blank form.



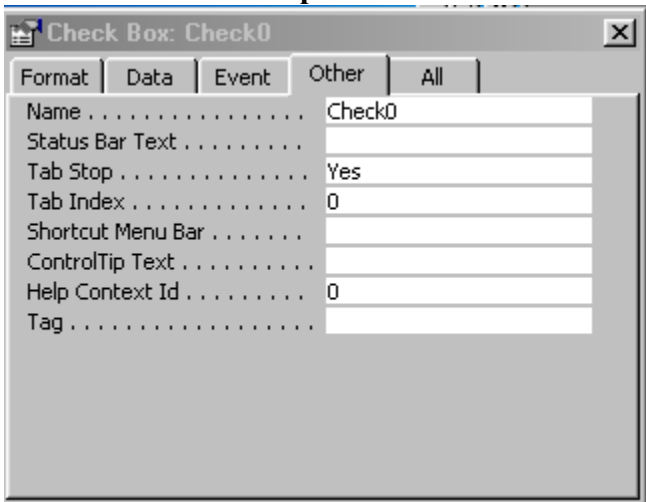
Double click **Create form in Design View**

This will give us a blank form with which to work. Add in 4 check boxes and labels for the 4 group categories listed in our table. Click the  symbol on the main toolbar and then drop the check box and label into the form. Do this three more times. The form should look like this when done:





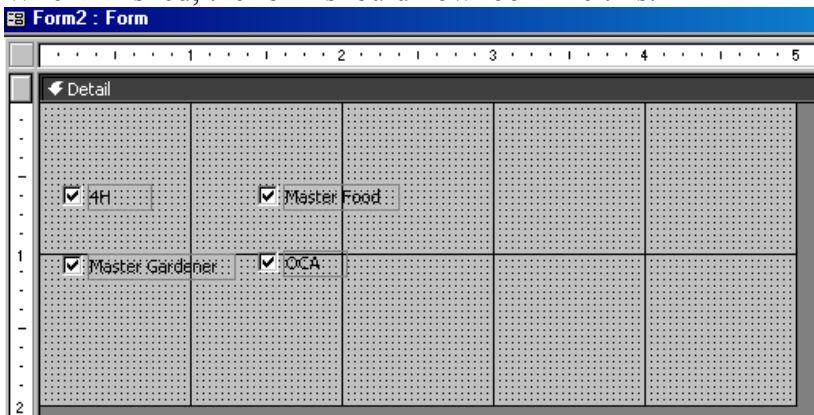
Next we need to change the names and labels of the check boxes. Right click the check mark for **Check0** and select **Properties**.




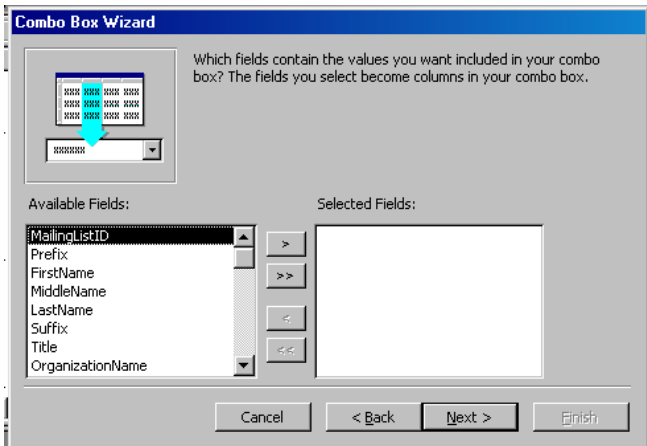
Click on the **Other** tab and change the **Name** field to **4H**. Click the **Data** tab and change the **Default Value** to **No**. Without leaving the properties area, left click the actual label **Check0**. Click the **Format** tab and then change its **Caption** property to **4H** as well.

Repeat these steps changing the **Check2** box to **Master Gardener**, **Check4** to **Master Food**, and **Check6** to **OCA**.

When finished, the form should now look like this:

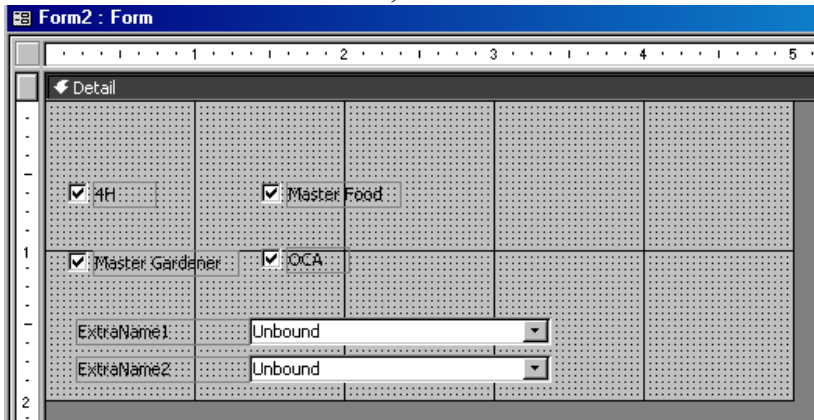


Next we need to add some blanks for you to select extra names. Quite often you send out mailings that you want to include one or two people that are not actually part of the group in question. Here is how to select these names. Click the  icon to select a combo box. Drop it in the bottom part of the form. The first question asked by the combo box wizard is whether we want it to look up values in a table. We do; click **Next**. Then it will ask what table to use. Select **Mailing List** and click **Next**.



Next we need to select the data fields that need to be displayed in the combo box. Click on **MailingListID** and then click the top > arrow. Repeat this step for **FirstName**, **MiddleName**, and **LastName**, then click **Next**. At the next prompt, click **Next** again. This question is just asking us whether we want to hide the **Key** field. What this means is that only the names will be displayed, but the record number will be the key data field. This will make more sense when we design our query.

When prompted what to name the combo box, type in **ExtraName1**, and then click **Finish**. You may need to resize the boxes for the label and combo box to fit the names. Right click the combo box, select **Properties**, click the **Other** tab, and change the **Name** field from **Combo#** to **ExtraName1**. Close the **Properties** window. Repeat these steps and name the second box **ExtraName2**. When finished, the form should look like this:

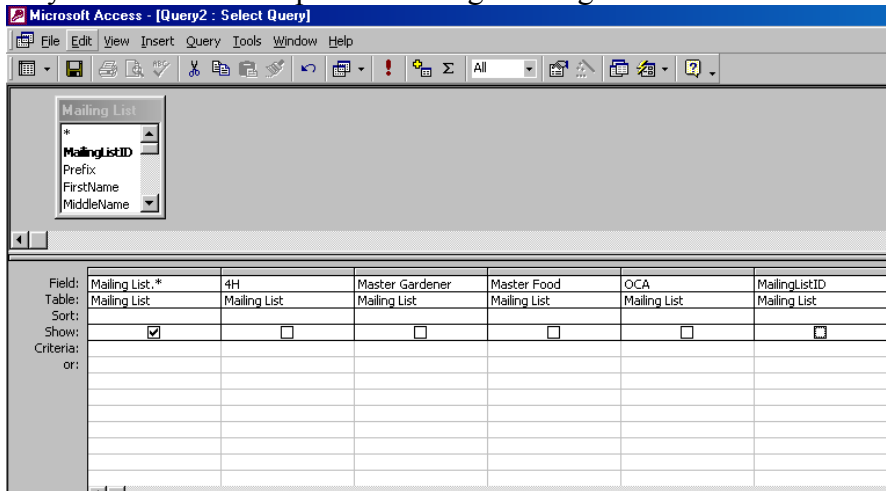


Close the form. Save it as **Form2** (you would name it something meaningful in real life). Double click the form to open it. Experiment with the ExtraName fields for a moment and see how they work. Leave the form open, but minimize it. We will need it open to create the Query.

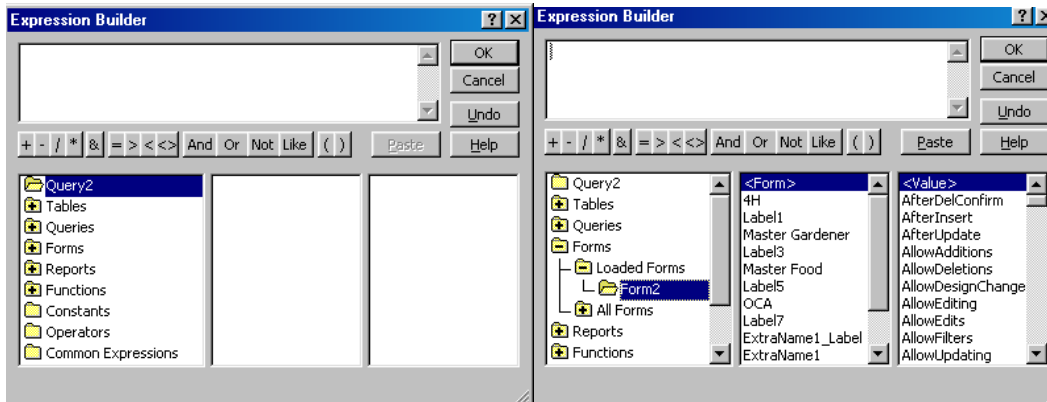


## Creating the Queries

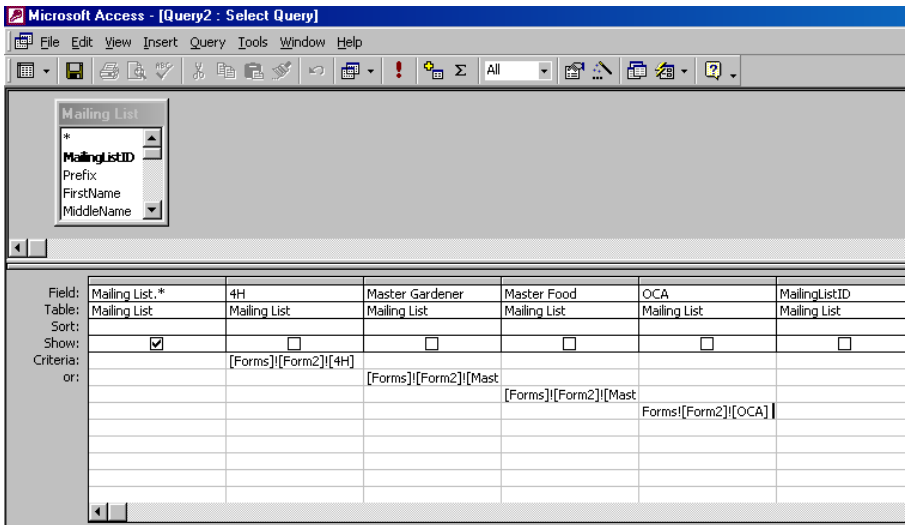
We left the form open to make it easy to find the fields when we build the query. Click on the query window, and then click **Create a Query in Design View**. The system will then ask you what table to use, and the only table **Mailing List** will be displayed. Click **Add** and then click **Close**. Set the first field to **Mailing List.\***. This means that no matter what match is made in our criteria, we want all the fields in the table to be available. This is always a good bet when you are not sure what fields you may wish to use in the future, and it allows us to use one query for multiple purposes. Now set up columns for **4H**, **Master Gardener**, **Master Food**, **OCA**, and **MailingListID**. Uncheck the **Show** option for these new columns. We already have these columns included in \* option, and it only wastes time and space including them again. The form should now look like this:



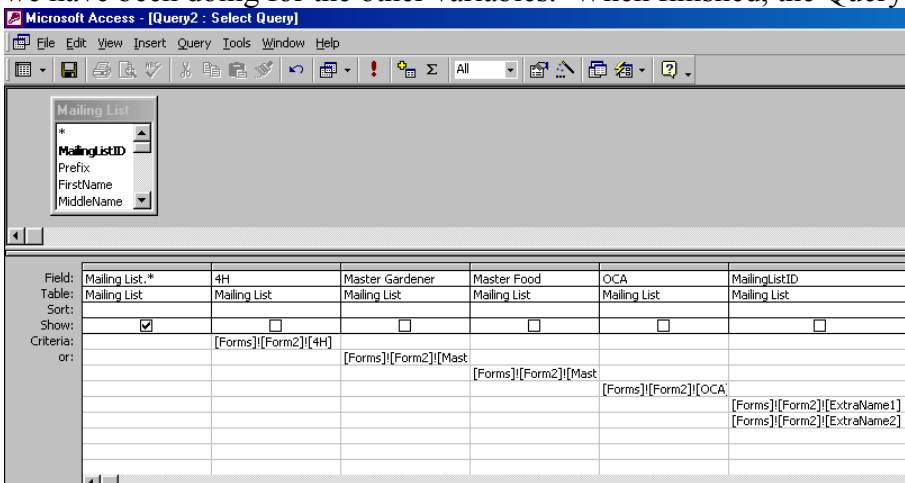
Now left click the **Criteria** box, right click, and then select **Build**. This brings up the expression builder.



Double click the **Forms** option, double click **Loaded Forms**, and then double click **Form2**. You will see the list for all the variables on **Form2**. Double click **4H** and then click **OK**. The criteria from the form will now be listed for **4H**. We need to add **and Yes** on to each line to insure that not only does the database agree with the form, but the category is checked as well. Now click the one line below the criteria block in the **Master Gardener** column, and repeat the build step selecting **Master Gardener** from the list. Move another line down, and repeat the steps for **Master Food**, and then **OCA**. When finished with these steps, your form will look like the picture on the following page.



Now we need to add the two ExtraName fields under MailingListID. Put them on separate lines like we have been doing for the other variables. When finished, the Query will look like this:



Close and save the Query as Query2. In real life, you would name your query something meaningful to your project.

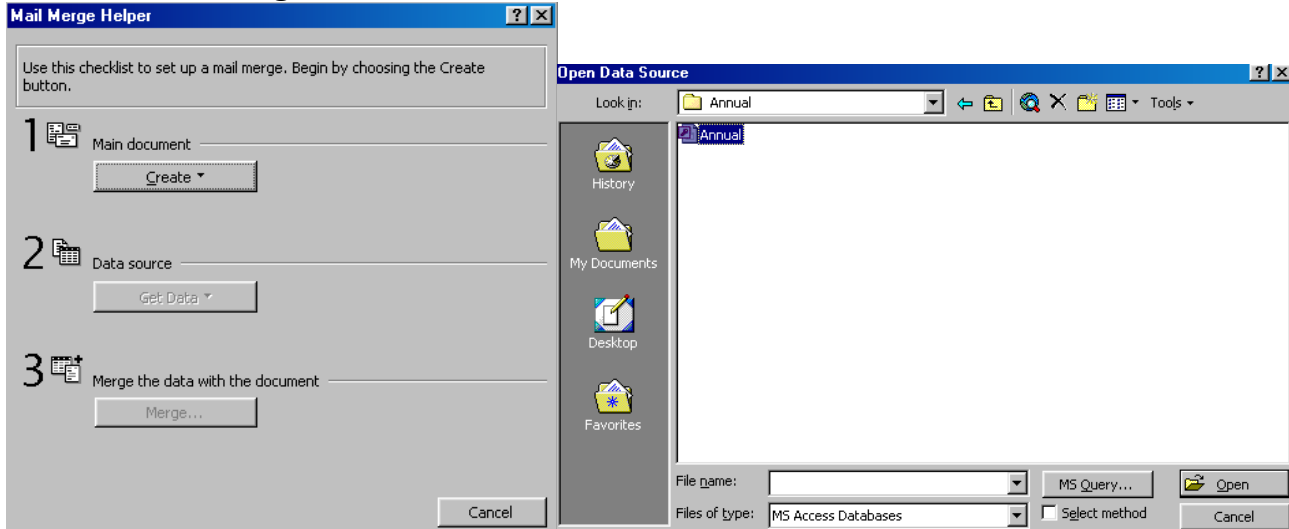
### Moving on to Word

Now that we have our form and query, we need to design the mailing labels and form letters in Word. Why use Word? Access has its own Mailing Label feature and report function, so why not use it? Word is far more flexible. When we design the actual mailing label, you will see that Word allows us to quickly change the styles of the labels and number of lines they contain where Access required rebuilding parts of the database to make a change. Also, Word allows for inputs and changes to the form letter before the mail merge which is not an option of any report within Access. Let's start with the Mailing Labels.

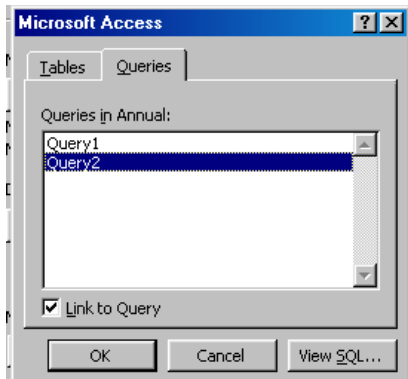


## Mailing Labels

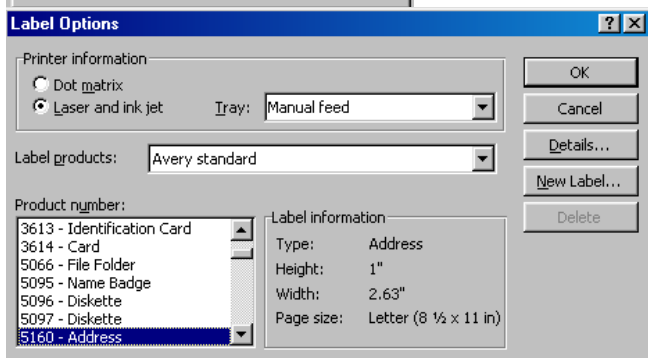
Before we start Word, pull up **Form2** and select a couple of categories (4H, OCA, etc.) so that we will have data to build sample labels. Then start Word. When the blank document loads, select **Tools and Mail Merge**.



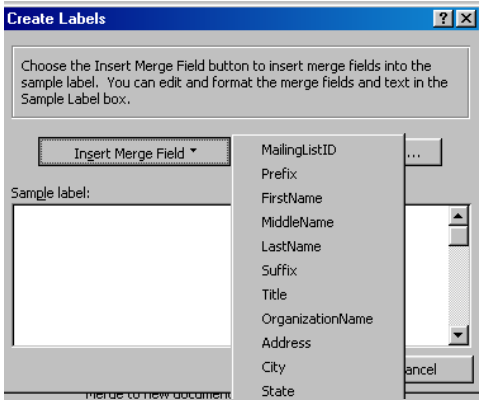
Click **Create**, select **Mailing Labels**, and then select **Active Window**. Next click **Get Data**, **Open Data Source**, select **Files of Type – MS Access Databases**, find **Annual**, and click **Open**. Remember to note where Access stores your database when you initially save, and what the exact name is. Many people assume when they first start using Access that the primary table name and the database name are the same. While you can do this on purpose, it is not done by default.



Select **Queries**, and then select **Query2** and click **OK**. Then click **Setup Main Document**.

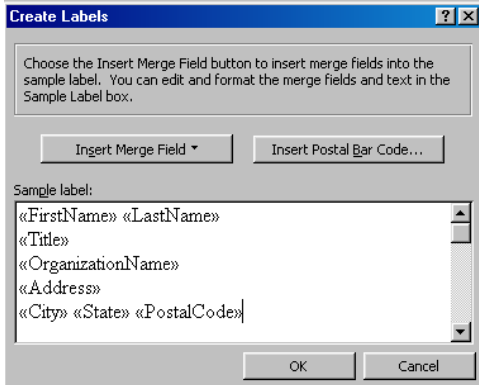


First we select the label type that we intend to print. For this exercise we will use Avery 5160 labels. Click **OK**,



Next we design the field layout for the labels. Click **Insert Merge Field**, select **FirstName**, hit space, click **Insert Merge Field** again, click **LastName**, and then hit **Enter**. This is how you add fields to the label. Now follow these same steps to add the following fields in this format:

Title  
OrganizationName  
Address  
City State PostalCode



When finished, the label will look like this. Click **OK**. Finally click on **Close**.

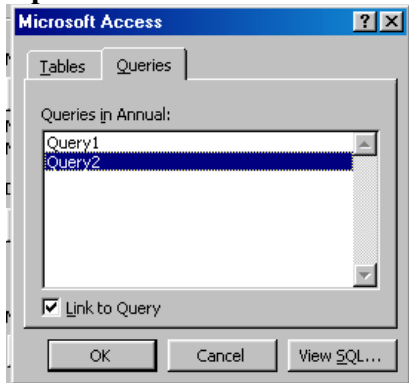
«FirstName» «LastName» «Title» «OrganizationName» «Address» «City» «State» «PostalCode»	«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»	«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»
«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»	«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»	«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»
«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»	«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»	«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»
«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»	«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»	«Next Record»«FirstName» «LastName» «Title» «OrganizationName» «Address»

The merge template will now load with all of our fields in place. Even though some of the fields do not show on this template, they will be listed if we run a merge. Save this file in the same directory as our database and call in MailingLabel2DOC. Finally close this document.

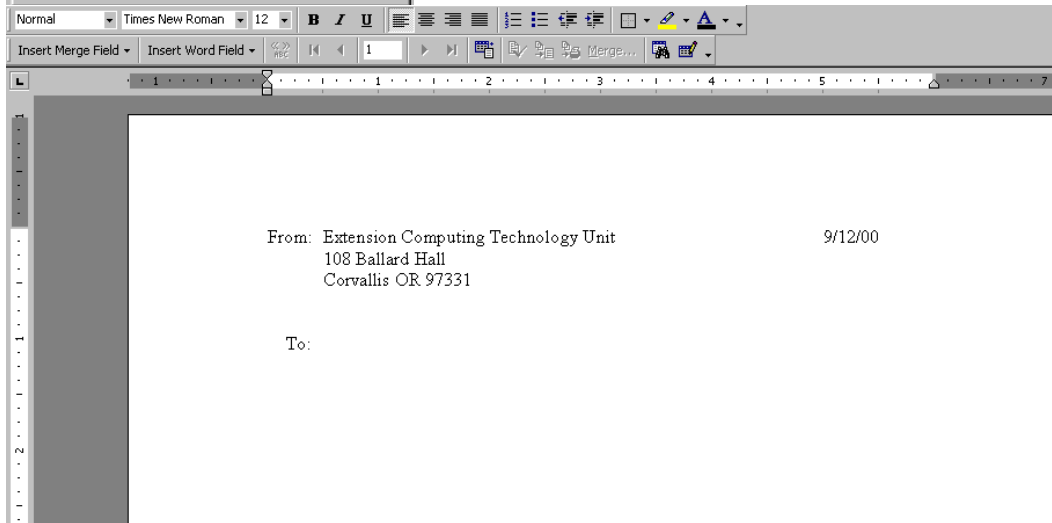


## Form Letters

Form letters are designed in largely the same manner as mailing labels. Open a new blank document in word. Click **Tools, Mail Merge, Create, Form Letter, and Active Window**. Next click **Get Data, Open Data Source**, select **Files of Type – MS Access Databases**, find **Annual**, and click **Open**.



Select **Queries**, and then select **Query2** and click **OK**. Then click **Edit Main Document**.



You start off with a blank page. Insert your **From** (if desired), the **Date**, and a **To:** place holder. These will not change and will be a part of every form letter. Next click **Insert Merge Field**. The selection process is identical to setting up a mailing label. Set up the following **To** block:

FirstName LastName Suffix  
Title  
OrganizationName  
Address  
City State Postal Code

Then put in a blank line, followed by **Dear** (a space), and then **FirstName LastName, .** Lastly add a couple of blank lines and then the statement: **Insert Text here**.



When finished, it should look like this:

```
From: Extension Computing Technology Unit          9/12/00
      108 Ballard Hall
      Corvallis OR 97331

To: «FirstName» «MiddleName»
     «Title»
     «OrganizationName»
     «Address»
     «City» «State» «PostalCode»

Dear «FirstName» «LastName»,

Insert Text Here|
```

Save this file as **FormLetter2** and then close this window

### Creating Buttons on the Query Form


The next step in creating our form is to create the buttons that will activate the mailing label and form letter templates. It is important that you know how to find the DOS name of a file in order to properly fill out the required fields in this section. If you right click a file and select properties, the **MSDOS Name** will be displayed. This will allow you to call this file from inside Access. For example, the FormLetter2.Doc should have the following Dosname:

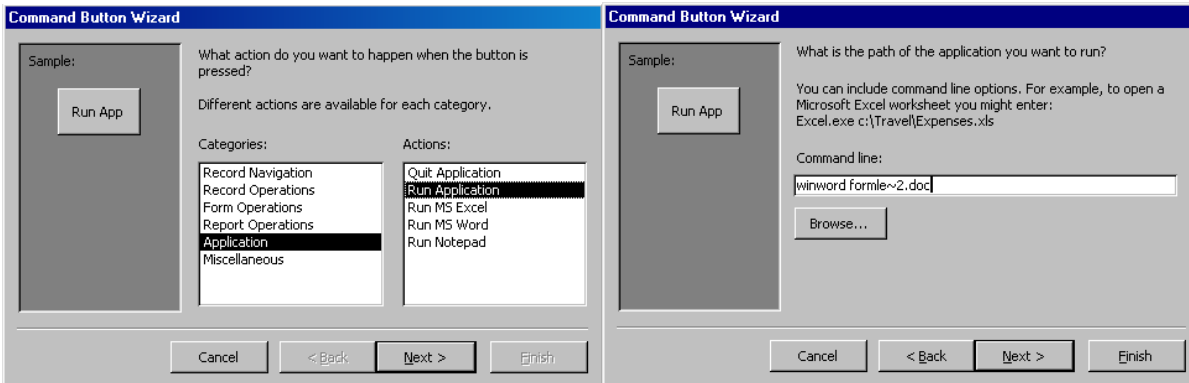
FormLe~2.doc

The DOS name of the mailing label file should be:

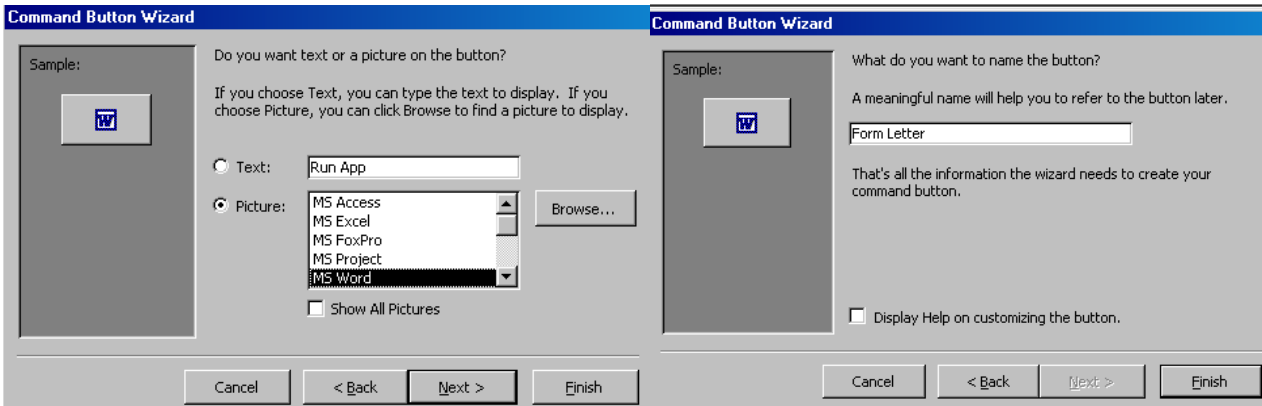
Mailin~2.doc

Creating the buttons is quite simple once you have this filename. Open **Form2** in **Design View**.


Click the Command Button icon , and then click the top part of the form to start a command button.



Select **Application**, **Run Application**, and then click **Next**. Type in **Winword formle~2.doc** and then click **Next**.



Select **Picture**, **MS Word**, and then click **Next**. Name the button **Form Letter**, and then click

**Finish**. Use the label tool  to put a text label next to the Form Letter button (such as Form Letter). Repeat the process for the **Mailing Label** template. When finished, your form should look like this:



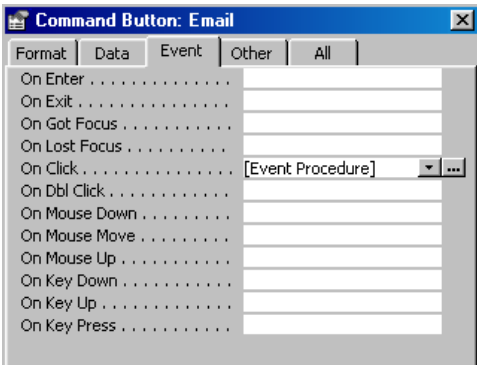


The form is now fully functional for mailing labels and form letters. Now we need to cover the interfaces into Outlook.

### Outlook Interfaces

Rather than try and teach the Visual Basic interface from scratch, included in this database are two files. The **Emailform** is a form that asks for a **Subject** line, **Body Text** field, and an optional **Attachment**. This form can be used in any program, and looks for a table named **EmailTable** that must have a field named **EmailAddress**. (This interface will not work unless the Outlook reference is loaded.) Also included is a query named **Emailquery**. This query creates the **EmailTable**. Copy the **EmailForm** to **EmailForm2** and the **Emailquery** to **Emailquery2**. Open the new **Emailquery2** in design view. If you look at each of the criteria fields, they are keyed to **Form1**, change the references to **Form2**. That is all we need to do to **Emailquery2**. Close and save this query.

We do not need to make any changes to **EmailForm2**. It is designed to look for **EmailTable**. All we have to do now is create the button for E-mail on **Form2**. The easiest way to do this is to copy the button and its visual basic code from **Form1**. Open **Form1** and **Form2** in design view. Copy the E-mail button and label from **Form1** to **Form2** and position it as desired. This only copies the button and the label. It does not copy the actual Visual Basic code that makes the button run. Go to Form one and Double click the Email button.



This will open a dialog box showing the Events listed for this button. An Event is what you want the system to do when the user performs a specific action. Click on the line next to **Event Procedure**. An arrow and three dots will appear. Click the three dots.

ControlType	104
Default	False
DisplayWhen	0
Enabled	True
EventProcPrefix	Email
FontBold	0
FontItalic	False
FontName	Tahoma
FontSize	8
FontUnderline	False
FontWeight	400
ForeColor	-2147483630
Height	576
HelpContextId	0

```
Private Sub Email_Click()

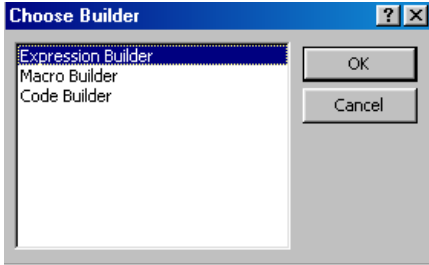
    Dim stDocName As String
    Dim stLinkCriteria As String
        stDocName = "EmailQuery"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    stDocName = "Emailform"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

End Sub
```

This will bring up all the Visual Basic code for our form. Highlight everything starting with the first **Dim** line and ending the second **DoCmd** line. Copy this text. You can now close **Form1**.



Go to **Form2** and double click the **Email** button. Click the blank line next to **On Click** in the Event list, and then click the three dots.



The system then asks what you wish to use to design the Event. Select **Code Builder** and click **OK**

```
Private Sub Email_Click()  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "EmailQuery"  
    DoCmd.OpenQuery stDocName, acNormal, acEdit  
    stDocName = "Emailform"  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
|  
End Sub
```

Paste the copied lines of code between the **Private** and the **End** statements as seen to the left. Change **EmailQuery** to **EmailQuery2** and **Emailform** to **Emailform2**.

Close and save the Visual Basic editor and **Form2**.

**Form2** is now ready to send e-mail as well.

These forms and the Visual Basic code they contain can be tailored to the needs of almost any database. For further help and or specifics, feel free to contact me. This is the first of many advanced subjects. If you have any suggestions for future subjects feel free to send them to me.

Kevin E. Presley  
ECTU Training Coordinator  
Oregon State University  
108 Ballard Hall  
(541) 737-0697  
[Kevin.Presley@orst.edu](mailto:Kevin.Presley@orst.edu)