

Access 2010: Buttons and macros

Practical workbook

Aims and Learning Objectives

By the end of this course you will be able to:

- create buttons to automate your database;
- create simple macros;
- create macros that do different things depending on the situation;
- create message boxes using macros.

Document information

Course files

This document and all its associated practice files are available on the web. To find these, go to <http://www.bristol.ac.uk/it-services/learning/resources> and in the **Keyword** box, type the document code given in brackets at the top of this page.

Related documentation

Other related documents are available from the web at:

<http://www.bristol.ac.uk/it-services/learning/resources>

Introduction

You can add buttons to Access data entry forms that, when clicked, perform one or more specified actions. You can create buttons either by using the Button Wizard, though there are limited options using this, or by building a macro and then creating a button that you press to run the macro.

Okay, so the next question is what is a macro? Macros consist of an action, or a list of actions, designed to perform, automatically, one or more tasks, such as opening a form, printing a report or finding a subset of records based on specific criteria.

In short, buttons and macros are about automating your database. However, you don't have to manually press a button to run a specific macro and can instead tell Access when to run a macro based on a specific event, such as tabbing into or out of a field or on entering data that matches specific criteria.

Prerequisites

You will get more out of this document if you have done one of the following:

- Attended “Access 2010: An introduction” (ITS-ACC10-1) and “Access 2010: Building Access databases” (ITS-ACC10-2).
- Worked through course documents for the above two courses.

Contents


Task 1	Simple automation using the Command Button Wizard	2
Task 2	Create simple macros and add command buttons to work them	3
Task 3	Create a conditional macro that runs automatically	5
Task 4	Automatically enter data and control navigation on a form	7
Task 5	Create custom message boxes	9

Task 1 Simple automation using the Command Button Wizard

Objectives To add simple command buttons to the Switchboard.

Comments When using the toolbox to create a button, make sure that the Use Control Wizards option is selected before you start (click on the Controls' more options dropdown in the Ribbon).

1.1 To create some simple buttons:

- Open the Buttons and macros database, **ButtonsAndMacros.accdb** – if in the training room, you will be shown where this is. If using this document on your own, you need to have downloaded the associated practice file and unzipped the database from there.
- Open **FrmSwitchboard** in **Design View**.
- In the **Controls** section of the **Ribbon** click once on the **Button** icon. 
- Click on the empty form to start the **Command Button Wizard**.

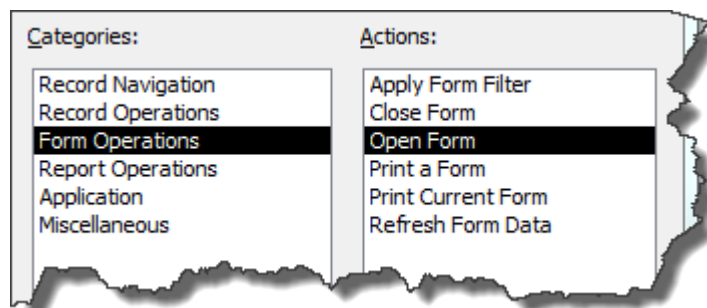


Figure 1 – you're going to create a button to open a form

- Select **Form Operations** followed by **Open Form**, then click on **Next**.
 - Select **FrmPatients** from the list and click on **Next**, followed by **Next** again.
 - Alter the **Text** caption to read **Open the patients form**, then click **Next**.
 - Name the button as **ButtonOpenFrmPatients** and click on **Finish**.
 - Move the button where you want it on the **Switchboard** form, then go to **Form View** and test the button.
- ### 1.2 To practice using the Command Button Wizard:
- On **FrmSwitchboard** create another button that opens **FrmDrugs** and another that opens the report **RptPatientLabels**.
 - On **FrmPatients** and **FrmDrugs**, create buttons that close each form.
- ### 1.3 To fine tune button properties:
- Right-click on the **Open the patients form** button and select **Properties**.
 - In **ControlTip Text**, enter **Open the patient form** – this will show a pop-up tip when you hover the mouse over the button.
 - Go to **Form View** and hover your mouse over the button.
 - Add pop-up tips for any other buttons you have already created.

Task 2 Create simple macros and add command buttons to work them

Objectives To create macros to open a specific object when the file is opened, to open a table as read only, to open a table showing specific records (using a conditional clause) and to assign command buttons to the last two macros.

Comments Apart from using command buttons, you can run macros from the Navigation Pane or by telling Access to run them based on specific events, much more of which later.

2.1 To create an **Autoexec** macro to open the **Switchboard** on file open:

- Click on the **Create** tab in the **Ribbon** and select **Macro**.
- In the box, select **OpenForm** from the dropdown list.

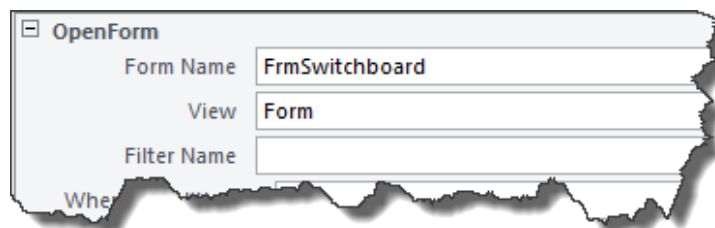


Figure 2 – creating a simple autoexec macro

- In **Form Name** select **FrmSwitchboard** and then close the macro, saving it as **Autoexec**.

Note When you open a database file, Access always looks to see if there is a macro called 'autoexec' and runs this first.

- Shut down the database and then reopen it – Access should automatically open the Switchboard for you - if not you probably spelled "Autoexec" wrong!

2.2 To create a macro to open a table as read-only:

- Click on the **Create** tab in the **Ribbon** and select **Macro**.
- Select **OpenTable** from the dropdown and in **Table Name** select **TblDrugs**.
- Leave the **View** as **Datasheet** and in **Data Mode** select **Read Only**.
- In the **Add New Action** box, select **Comment** from the dropdown and enter ***This macro opens TblDrugs in read-only mode.***
- Click on the **Save** icon and save the macro as **MacroTblDrugsReadOnly**.

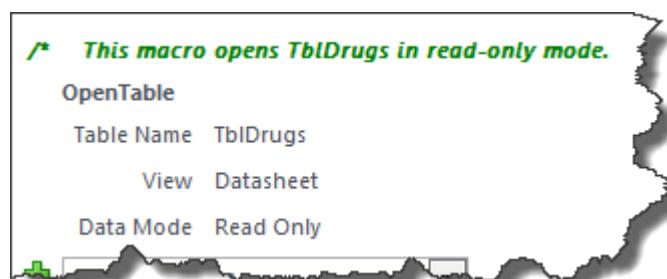


Figure 3 – move the comment to the top

- Hover the mouse over the comment and click on the upward, green arrow that appears to the right to move the comment to the top.

Note It is a good idea to add comments because macros can be very complex, so giving yourself an idea of what the macro does and how it does it is a good idea – don't kid yourself that you will always remember how you built a complex macro!

- Save the macro again then close it and run it from the **Navigation Pane**.
- Try to alter some of the data just to make sure you chose the right options.

2.3 To open a form to show specific records based on a data value:

- Create a new macro and select **OpenForm** from the dropdown.
- In **Form Name** select **FrmPatients** and leave **View** as **Form**.
- In **Where** condition enter **[Gender] = "female"**.
- Save the macro as **MacroFrmPatientsFemaleOnly**, close it then run it from the **Navigation Pane**.
- Use the navigation buttons to see that only female patients are shown.

2.4 To create command buttons to run the last two macros:

- Close any open objects then open **FrmSwitchboard** in **Design View**.
- From the **Controls** section of the **Ribbon** click on the **Button** icon then click once on the form to start the **Command Button Wizard**.
- In **Categories**, select **Miscellaneous** and in **Action** select **Run Macro**.
- Click on **Next**, select **MacroTblDrugsReadOnly** and click on **Next**.
- Select **Text** and enter **Open Drugs table as read-only** and click **Next**.
- Save the **Command Button** as **ButtonOpenTblDrugsReadOnly**.
- Do exactly the same as you've just done, but select the other macro, add appropriate button text and give the **Command Button** an appropriate name.

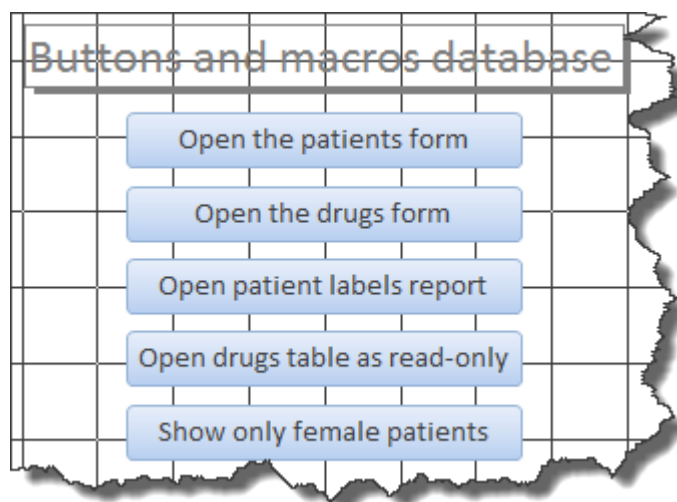


Figure 4 – the new buttons

- Size and format the buttons to taste, then switch to **Form View** and test your new buttons.

Task 3 Create a conditional macro that runs automatically

Objectives To create a macro that runs if data in a field matches specific criteria and set this macro to run automatically, without needing to press a button.

Comments You can add further conditions if you need to do multiple things depending on different values.

- 3.1** To create a conditional macro that changes the colour of the Gender field based on whether the person is female:
- Create a new macro, select **If** from the dropdown and enter **[Forms]![FrmPatients]![Gender]="female"**.
 - In the **Add New Action** field choose **SetProperty** from the dropdown.
 - In **Control Name** enter **Gender** and in **Property** choose **BackColor** from the dropdown.

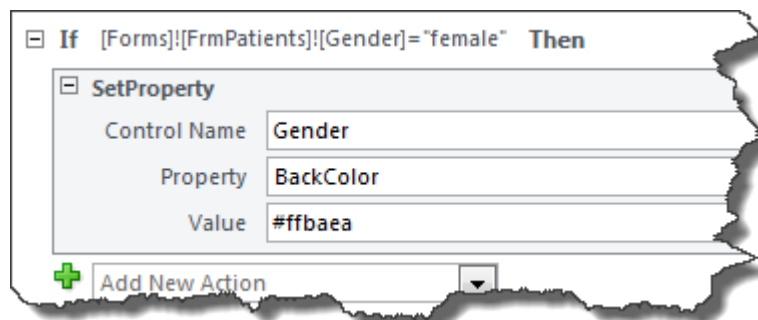


Figure 5 – how the macro should look

- In **Value**, enter **#ffbaea** (which is a nice pinky-mauve colour).
- 3.2** To assign the macro to a form event and test it:
- Save the macro as **MacroSetBackColour** and open **FrmPatients** in **Design View**.

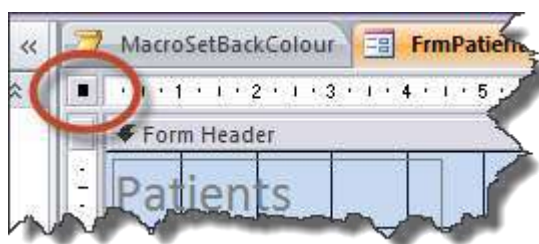


Figure 6 – click the circled area to view the overall form's properties

- Click on the box at the top, left-hand side of the form so that a small black square appears and then open the **Property Sheet**.
- Select the **Event** tab in the **Property Sheet** and in **On Current** select your new macro from the list.

Note Any macro attached to On Current runs both when a form is opened and whenever you move from one record to another

- Go to **Form View** and notice that the background has now changed.
 - Check a few records and notice that male patients have also changed colour – this is because we haven't yet told the macro what to do if the condition is not met.
- 3.3** To tell Access what to do if the person is male or if gender has not been disclosed:
- Click on the **Else If** link and enter **[Forms]![FrmPatients]![Gender]="male"**.
 - In the **Add New Action** field choose **SetProperty** from the dropdown.
 - In **Control Name** enter **Gender** and in **Property** choose **BackColor** from the dropdown.

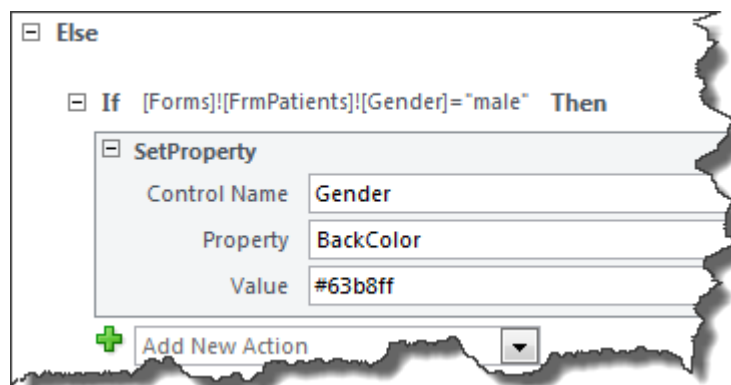


Figure 7 – how the next part of the macro should look

- In **Value**, enter **#63b8ff** (which is a nice light blue colour – not that we're doing any stereotyping here, or anything).
- Now click on the **Else** link and choose **SetProperty** from the dropdown.
- In **Control Name** enter **Gender** and in **Property** choose **BackColor** from the dropdown.

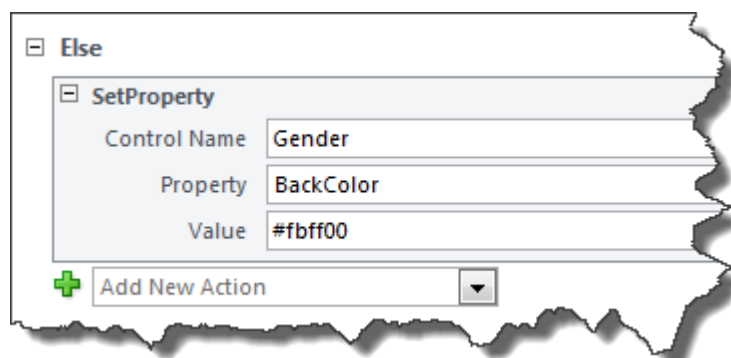


Figure 8 – how the last part of the macro should look

- In **Value**, enter **#fbff00** (a nice yellow), then save and close the macro.
- 3.4** To test the finished macro:
- Open **FrmPatients** and check through a few records.
 - Change the gender of one of the patients to **Undisclosed** and then click on the **Refresh** icon in the **Ribbon**.

Task 4 Automatically enter data and control navigation on a form

Objectives To automatically add set data to a specific field and tab to a specified field based on the value of data entered.

Comments In the following example, if you enter Bristol into the Town field, Access automatically adds Bristol to the County field (yes, Bristol has been officially returned to its earlier status as both a City and a County since the County of Avon was abolished in 1996) and tabs directly to the Postcode field.

Note MS classifies the SetValue macro command we're going to use as an 'Unsafe Action'. If you use Unsafe Actions in a macro, you may have to add the location of your database to Access' Trusted Locations list otherwise macros will be disabled. To find out how to add a location to your Trusted Locations, go to http://office.microsoft.com/en-gb/access-help/introduction-to-access-2010-security-HA010341741.aspx#_Toc265079112

4.1 To set the tab to go directly to Postcode if the town is Bristol:

- Create a new macro, select **If** from the dropdown and in **Conditional Expression** enter **[Forms]![FrmPatients]![Town]="bristol"**.

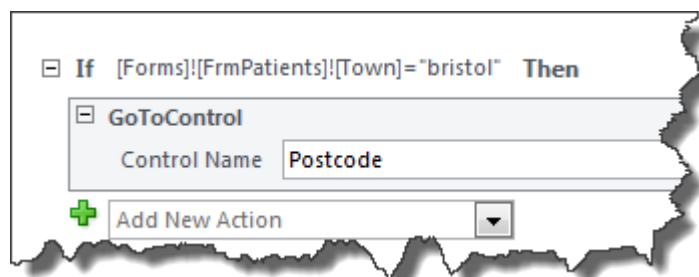


Figure 9 – how your macro should look at this point

- In the **Add New Action** box directly below the **If** select **GoToControl** and in the **Control Name** box enter **Postcode**.
- Save the macro as **MacroTownIsBristol** and open **FrmPatients** in **Design View**.
- Click on the **Town** field control and open the **Property Sheet**.

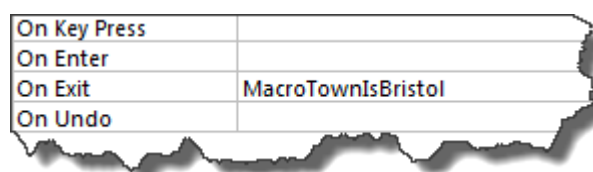


Figure 10 – select the new macro in the On Exit event

- Select the **Event** tab, click in **On Exit** and select your new macro from the dropdown list.
- Switch to **Form View**, go to a new record and start to enter details for a new patient from **Bristol** to test the macro.

4.2 To set Access to automatically fill in the County field when town is Bristol:

- Click on the **Action Catalogue** icon on the **Ribbon** if not already selected and then click on the **Show All Actions** icon as well.



Note The SetValue is hidden unless we specifically choose to use it, in which case it can only be selected when we select Show All Actions.

- In the **Actions** section of the **Action Catalogue** expand **Database Objects** and double-click on **SetValue**.
- In **Item** enter **[Forms]![FrmPatients]![County]** and in **Expression** enter **"Bristol"** (you must add the speech marks).

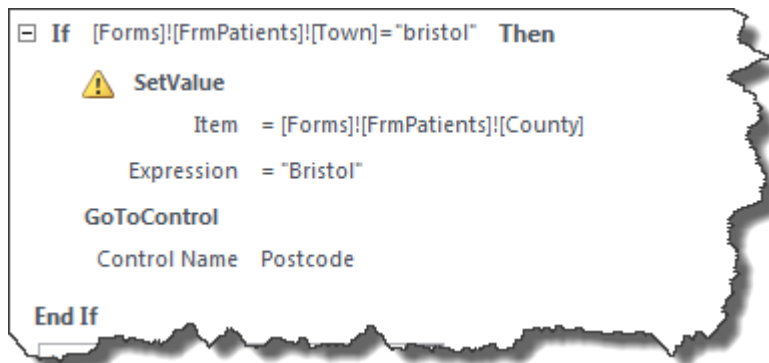


Figure 11 – how the macro should look when finished

- Now click on the green, upward pointing arrow to the right to move the **SetValue** command above the **GoToControl** command.
- Save the macro then test it out on **FrmPatient**.

Task 5 Create custom message boxes

Objectives To create a simple message box that gives a warning message, to create a message box that does a calculation based on data on a form and to create message box that displays different messages based on the data entered into a field.

Comments Access' own messages are not always user-friendly, so creating your own message boxes can be a good thing for the database's users!

5.1 To create a simple message box:

- Create a new macro and select **MessageBox** from the dropdown.
- In **Message** enter *Have you selected a valid dosage?@Is this the right drug?@Why not double-check just in case?*
- Set **Beep** to **No** – beeps can be very annoying – and set **Type** to **Warning!**

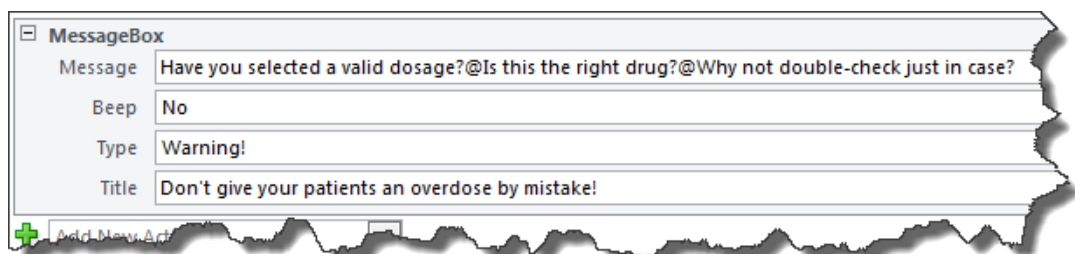


Figure 12 – how the macro should look when finished

- In **Title** enter *Don't give your patients an overdose by mistake!*
- Save the macro as **MacroDosageWarning** and open **FrmPatients** in **Design View**.
- In the subform, click on the **DosesPerDay** field and open the **Property Sheet**.
- In **After Update** add your new macro from the dropdown.
- Go to **Form View**, change one of the values in the **Daily dosage** field, save the record and look at the message to see what the @ symbols do.

5.2 To create a message box that makes a calculation based on data on a form:

- Create a new macro and select **MessageBox** from the dropdown.
- In **Message** enter *= "Each tablet costs £" & [Forms]![FrmDrugs]![CostPerPacket]/[Forms]![FrmDrugs]![TabletsPerPacket]*.
- Set **Beep** to **No** and set **Type** to **Information**.

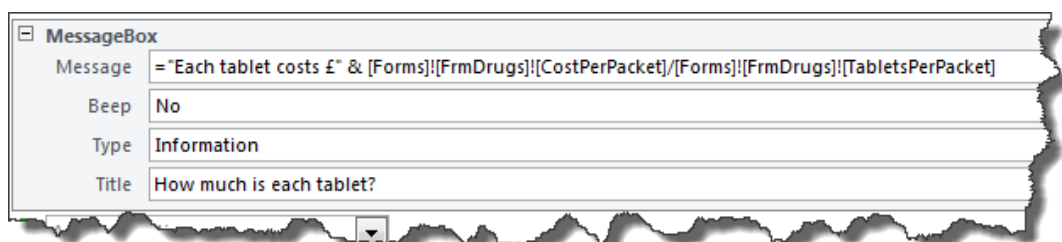


Figure 13 – how the macro should look when finished

- In **Title** enter *How much is each tablet?*
- Save the macro as *MacroCostPerTablet* and open **FrmDrugs** in **Design View**.
- Use the **Button Wizard** to add a new button that runs your new macro.

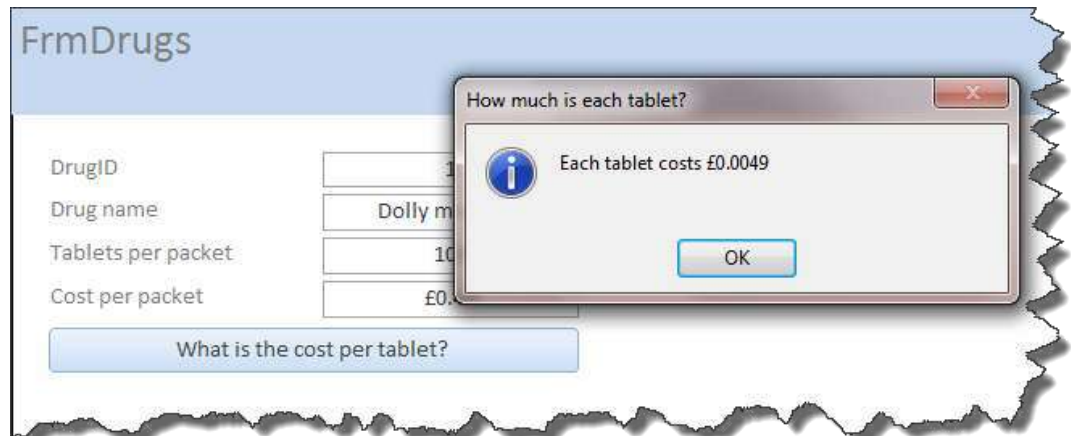


Figure 14 – this is what should happen ... if not, go back and check your macro!

- Go to **Form View** and test your new button and macro.
- 5.3** To create a message box that displays different messages based on the data entered into a field:
- Create a new macro, select **If** from the dropdown and enter *[Forms]![FrmPatients]![Country]="UK"*.
 - In the **Add New Action** box below select **MessageBox** from the dropdown.
 - In **Message** enter *International dialling code is +44*.
 - Set **Beep** to **No** and set **Type** to **Information**.

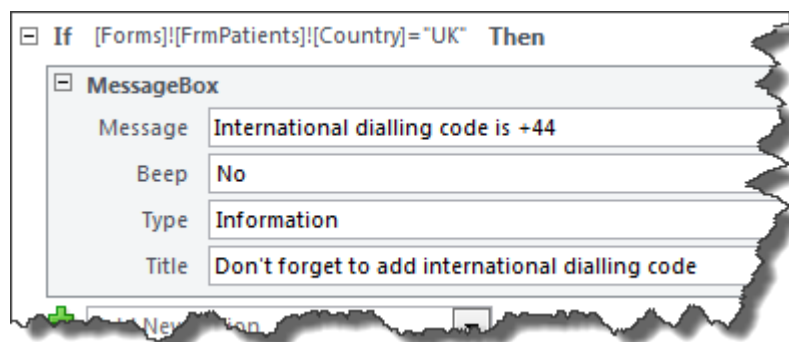


Figure 15 – how the macro should look at this point

- In **Title** enter *Don't forget to add international dialling code*.
- Save the macro as *MacroCostPerTablet* and open **FrmDrugs** in **Design View**.
- Save the macro as *MacroInternationalCode* and open **FrmPatients** in **Design View**.
- In the form click on the **Country** field and open the **Property Sheet**.
- In **After Update** add your new macro from the dropdown.

- Go to **Form View** and add **UK** to one of the records, then tab out of the field to test the macro so far.
- 5.4** To complete the macro:
- Click somewhere inside the current macro so that you can see and click on the **Else If** link.
 - In the **Else If** box enter **[Forms]![FrmPatients]![Country]="France"**.
 - In the **Add New Action** box below select **MessageBox** from the dropdown and do exactly as you did with the first message box, except make the dialling code **+33**.
 - Add another **Else If** and do as above but for **Germany** with the code **+49**.
 - Click on the **Else** link then select **MessageBox** from the dropdown.

```

If [Forms]![FrmPatients]![Country]="UK" Then
    MsgBox
    Message International dialling code is +44
    Beep No
    Type Information
    Title Don't forget to add international dialling code

Else If [Forms]![FrmPatients]![Country]="France" Then
    MsgBox
    Message International dialling code is +33
    Beep No
    Type Information
    Title Don't forget to add international dialling code

Else If [Forms]![FrmPatients]![Country]="Germany" Then
    MsgBox
    Message International dialling code is +49
    Beep No
    Type Information
    Title Don't forget to add international dialling code

Else
    MsgBox
    Message Unknown country - please add international dialling code
    Beep Yes
    Type None
    Title Don't forget to add international dialling code

End If

```

Figure 16 – how the finished macro should look

- In **Message** enter **Unknown country - please add international dialling code** and in **Title** add the same as for the previous messages. Do as you please with the **Beep** and **Type** options!
- Save the macro and go and try it out on **FrmPatients**. Okay, to be fully useful, you'd need an *Else If* for every likely country, but you get the idea!