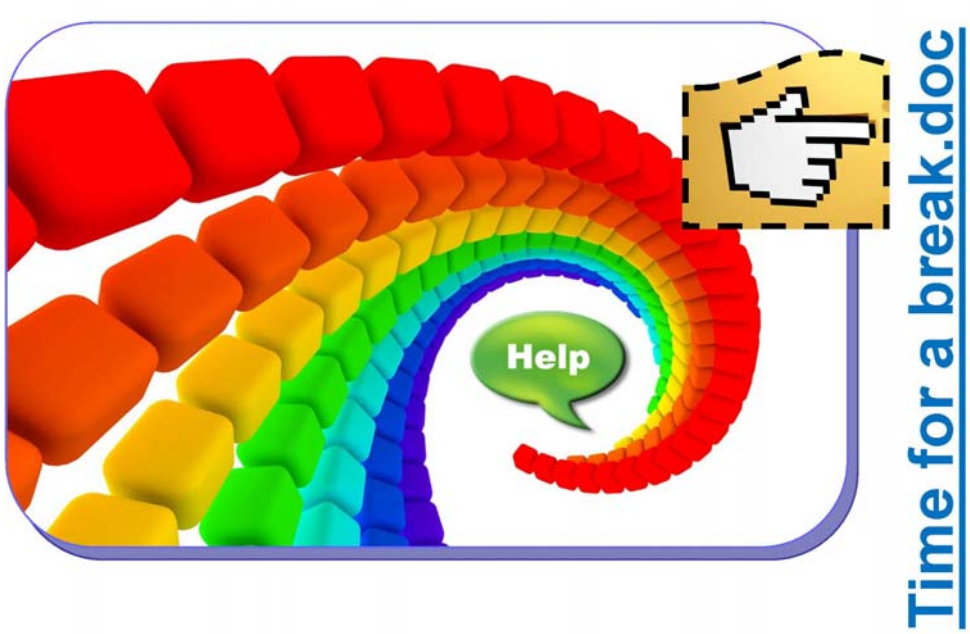
Chapter 7.

Buttons, Hyperlinks & Bubble Help



Buttons and hyperlinks are a pushover with Alpha Five – and so is Bubble Help…

Just a few simple steps and you will be creating all types of buttons and hyperlinks. They can open forms and browses, print reports and letters and run complex operations. Alpha Five lets you dress up buttons with built-in and custom graphics or keep them plain and simple. The choice is yours.

We'll start you on the road to some pretty exciting bubble help, too.

Overview

While the basic topics covered here are buttons and bubble help, the sub-topics are quite diverse. So we thought a chart of what to find where would be helpful when you design your own application. As always, exercises are created in "teaching order," so look around for what you need. Details are in the Table of Contents and Index. **V11 feature(s) in green.**

Don't miss out!

Learn how to use the HTML editor to give your forms Bubble Help with a bang. Here's a peek at a lesson in *Alpha Five Made Easy*

• Bubble help can keep your forms from becoming too crowded. Imagine using the customer's hobby to personalize a phone conversation.

a. *Control Panel > Forms* tab: Double click on **BubbleHelp** to open in View Mode.

b. Hover the cursor over the **First Name** field. We have set a delay, so it will take a moment to come up. (Not all records have hobbies entered, so page through the records.)

c. Put your cursor over the company, city and photo fields to get more ideas for **your** app!

Preparation for the Lesson.

Open Alpha Five and navigate to the following file:

- c:\A5_MadeEasyDesktopBook\ABC_DesktopLessons\ABC Seminars.adb*
-  Click on the Control Panel tab in the Window Bar at the bottom of the screen to bring the Control Panel to the front.

What's ahead in this chapter

PLEASE TAKE A MOMENT TO READ THIS

Buttons are like the icing on the cake. After all the hard work we've done so far, you'll have fun with this chapter. We'll create buttons that print a report we created earlier, run the series of operations we made in Chapter 6, add A-Z directory buttons to a form, and, coolest of all, open a web component without even putting it on a form.

In addition, we'll show you how to make a hyperlink without writing code, how to customize fly-over effects and learn button help basics.

This chapter also introduces another Alpha Five wonder, *Action Scripting*. While you follow a series of prompts, Action Scripting writes the Xbasic code in the background.

Sweet!

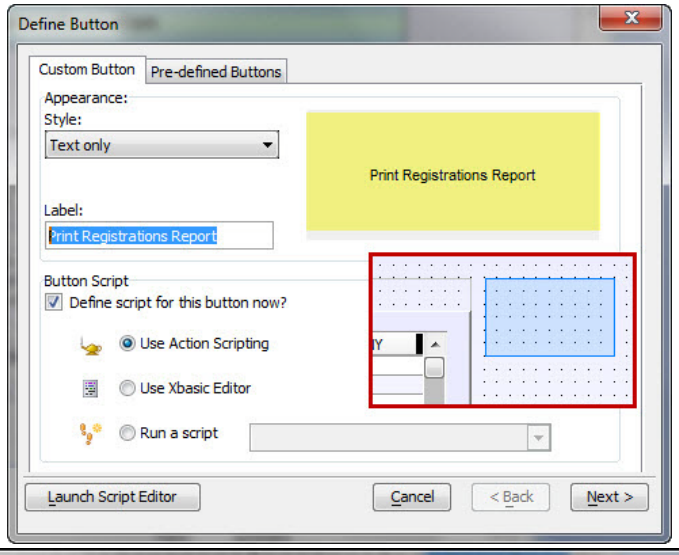
Creating a Button that Prints a Report

It certainly would be awkward for the end user to have to go to the control panel to search for the proper report in order to print it. Action Scripting makes it easy to run it with a button.

- This exercise will use the demo report, *RegistrationsFF*. If you completed Chapter 5, use *MyRegistrationsFF*.

1. Control Panel > Forms tab: Open **ClientInfo_Buttons** in Design Mode. (If you did the exercises in Chapter 4, open the *MyClientInfo* form.)
2. *Toolbox:* Click the Button control.

*. Depending on how Alpha Five is opened, the file extension **.adb** may not appear. (For instructions on opening an existing database, see "Opening an Existing Database" on page 14.)



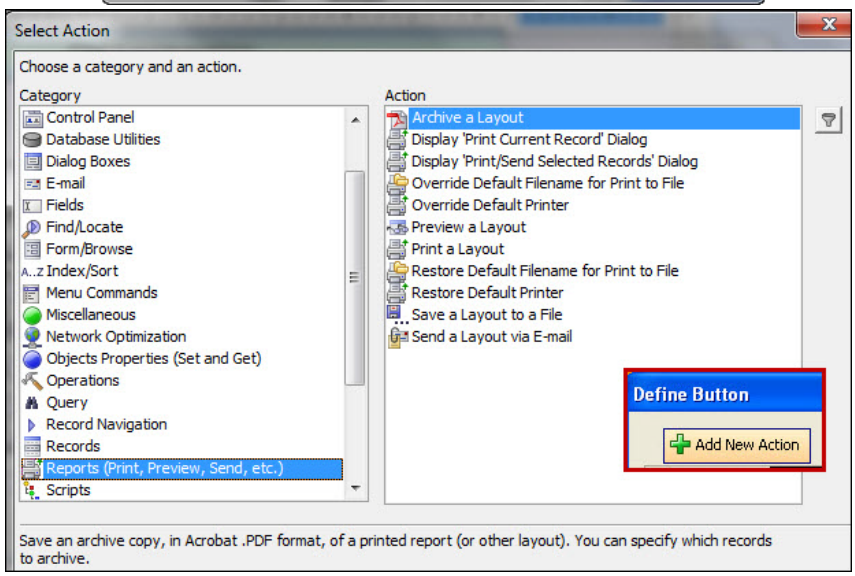
3. Draw a box on the blank area to the right of the Tabbed object (inset at left).

Dialog: Define Button

4. *Label:* Overwrite Script with **Print Registrations Report**.

- The preview window (yellow) shows how the button will look initially. Once the button is created, we will refine its appearance.
- *Style:* Buttons can have text and/or bitmaps (graphics). We will add them later at the Properties Pane instead.*

5. *Button Script:* Choose **Use Action Scripting**. (Click Next)



Using Action Scripting

Action Scripting writes the Xbasic code in the background, so all you have to do to is pick from the menus.

Dialog: Define Button (page 2)

6. Click **Add New Action** (inset) and use the following settings:

Dialog: Select Action

a. *Category*: Choose **Reports (Print, Preview, Send, etc.)**

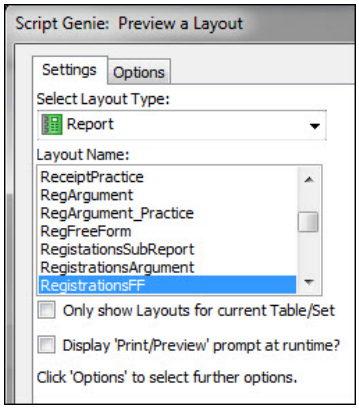
b. *Action*: Choose **Preview a Layout**. (Click OK)

PRINT VS PREVIEW

- **Print:** If there is no record, Alpha Five sends a message to the printer and a page is printed that says there is no record.
- **Preview:** If there is no record, the users is notified on screen.

We almost always use Preview because we like to know right away if there is an error. Admittedly, it is an extra step and there are times when going straight to the printer saves time. Also, there is another way to add the options (step 8 below).

*. If you want to use this window, choose the Style, click Define Picture, choose Internal and click the up arrow. Then follow from step a on page 301.



Dialog: Script Genie: Preview a Layout

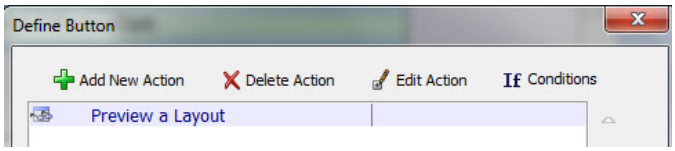
7. *Layout Name:* Choose **RegistrationsFF**.

8. *Display 'Print/Preview' at runtime?* **Yes.**

- This actually means: Display the *Print Method* dialog that has both Print and Preview radio buttons (screen shot at "Print Method dialog" on page 300).
  - If you choose Yes, the dialog at will appear.
  - Selecting No here will send the report straight to Preview or to the Windows Printer dialog.*

9. Click NEXT.

10. *Select Records to Print*: No additional selection criteria. (Click Finish)



- The Action Script appears in the Define Button dialog.
- There is one thing missing. If we came back later to see what report was printed, we would have to click Edit Action to look inside. From experience, we have learned its best to make a notation first.

Using the Script Editor

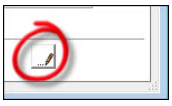
The Define Button dialog above has only an abbreviated editor, so we prefer to use the full Script Editor. The problem is that you can't activate it for use with Action Scripting until you have entered the first one.

The main reason we like it better is because the action can be identified as you go along. Trust me, you *will* need to return six months from now to review a script. If you have entered the name of the report or form, etc., your job will be much easier.

Dialog: Define Button

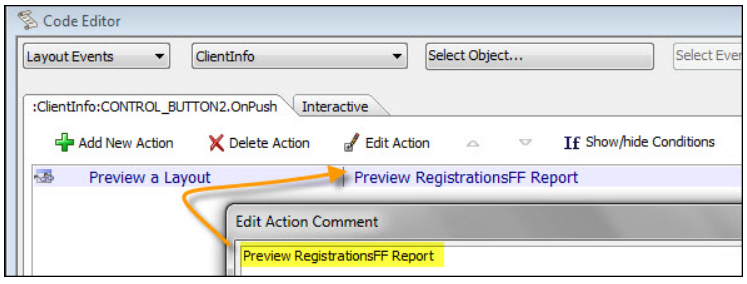
11. Click the **Launch Script Editor** button at the bottom of the screen.

CODE EDITOR



Dialog: Code Editor

12. Click on the teeny tiny pencil at the bottom right of the screen to open Edit Action Comment (yellow below).



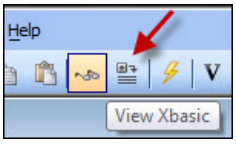
Dialog: Edit Action Comment

a. Type: **Preview RegistrationsFF report.**

b. Click OK to put the comment in the Code Editor.

Since we're here, let's look at the code – this is the Code Editor, after all.

VIEW / CONVERT TO XBASIC



There are two buttons on the toolbar that (1) View the Xbasic created in the background by the Action Scripting genie (orange at left) and (2) Convert the script to Xbasic (arrow).

*. To bypass the Windows Printer dialog, see page 257.

In the former, you can copy all or part of the code as usual. In the second, the code is actually converted (can't be undone).

13. Click **View Xbasic**. Take a look and then click Close.

FUTURE XBASIC USERS

Are you beginning to see how the code is developed?*

NON-PROGRAMMERS

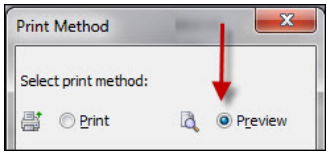
Now see, that wasn't so scary after all!

14. CTRL + S to save the Action Script; close the Code Editor.
    - The button appears.
15. CTRL + S to save the form changes; go to Form View.

- We saved the code, but the form must also be saved in order for the changes to hold. It is not necessary to do it each time you go to Form View, but it is a good practice.

16. Click the button to test it out.

PRINT METHOD DIALOG



- In step 8 we told Alpha Five to display the *Print Method* dialog. The radio button is set at Preview because we selected **Preview a Layout** in step b on page 298. Had we chosen to **Print a Layout** at that time, the *Print* radio button would be set as the default.

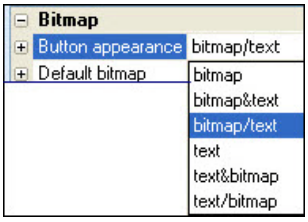
17. Click OK to preview the report.

18. Look it over and then click **Exit Preview** to close the report.
19. At the form, return to Design Mode.

Enhancing the appearance of a button

Next we will add a graphic (aka bitmap) and change the background and font color of the button. Then we'll add bubble help to give the user additional information.

20. Click on the button to select it.



BITMAP / TEXT OPTIONS

You can choose to show only a bitmap (graphic) or only text or a combination of the two. There are several appearance choices. (If necessary, stretch the right side of the pane to make it wider so you can see them.)

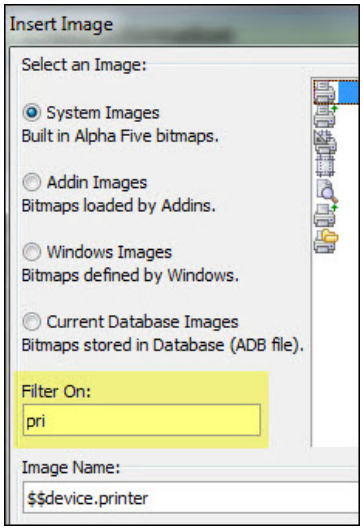
- Bitmap & text puts them side-by-side.
- Bitmap/Text puts the graphic above of the text.

PROPERTIES PANE: CONTROL_BUTTON1

BITMAP

- *Button appearance:* **Bitmap/Text**.
- *Default bitmap:* (see below)

*. A good time for a plug for Dr. Peter Wayne's *Xbasic for Everyone* book. Available at www.libertymanuals.com.



a. Default bitmap: Click the button.

Dialog: Insert Image

Alpha Five has a built-in library of images that you can use on buttons.

b. *System Images*, *Addin Images* and *Windows Images*. Click each to see their images.

- *Current database images:* You can develop your own gallery of bitmaps (see "Adding Custom Graphics for your Buttons" on page 311).

QUICK PICK

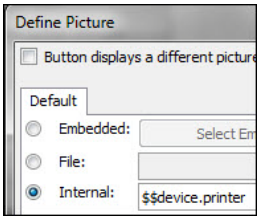
Since the library list is so long, it could take a while to find the right symbol. We can shortcut the process by entering a filter that will reduce the selection.

c. Click *System Images* (default).

d. *Filter On:* To find graphics suitable for printing, type **pri** as at left.

e. Choose the top icon or any other to enter the image name in the bottom box.

f. Click INSERT.



The name will no longer be available at the Properties Pane, but you can see it at Object Properties.

OBJECT PROPERTIES (F12): SETUP TAB:\

a. Click Define button bitmaps.

b. Cancel out of Object Properties.

PROPERTIES PANE: CONTROL_BUTTON1 (CONT.)

FILL

- *Accent color:* **Pale blue.**
- *Color:* **Turquoise.**
- *Style:* **Gradient Horizontal**.
  - Notice that even though we used the same colors as the header, the effect is slightly different because we used a different gradient style.

FONT

- *Bold:* **Yes.**

HELP

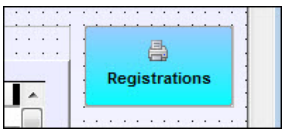
- *Bubble Help:* **Preview/Print Registrations Report.**
  - Bubble help is discussed in detail on page 326.

MISC

- *Text:* **Registrations.**
  - Since we now have a printer icon and bubble help, we can shorten the button name from *Print Registrations Report* to *Registrations*.

Adjusting button placement on the form

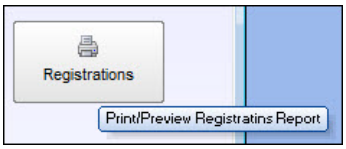
The process for moving, sizing and aligning buttons is the same as for other objects on a form: You can also evenly space a series of buttons by using the *Arrange Toolbar* and copy formatting from one to another with the *Copy Appearance* toolbar



21. Resize the button so it is just large enough for both graphic and label.
22. Align it with the top of the Tabbed Object.
23. Save the form and go to Form View.
    - The button takes on the appearance of the settings we applied.

24. Move the cursor over the button (hover) to see the Bubble Help.
    - Notice that hovering turns the button white, We will customize that effect next.

25. Return to Design Mode.



Customizing a fly-over effect

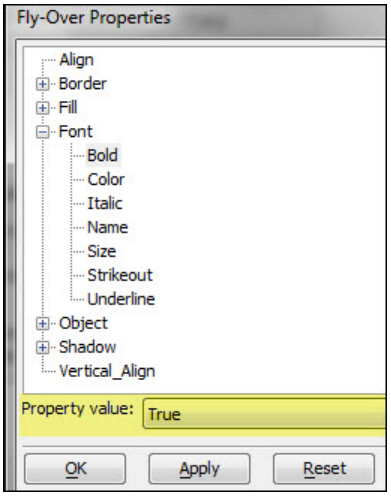
The effect that occurs when you hover (move your cursor) over the button, is known as *Fly-Over*. By default, new buttons have it set to white with black text as above. That can easily be changed – or it can be turned off completely.

There won't be any changes to see at design mode, you have to go to Form View to see the effect.

OBJECT PROPERTIES (F12): CONTROL_BUTTON1

SETUP

- *Fly-over Effects:* **Yes**.
- *Set Fly-over Appearance:* Click the button.



The settings can be made in a batch or you can click Apply after each one. Changes are made in the *Property value* box at the bottom.

- **Apply button:** Saves current edits.
- **Reset button:** Reverts an edit back to the default. In other words, if you set Bold = True, click Apply and then click Reset, it will return to False.
- **OK button:** Saves all edits and closes the dialog.

c. Choose the following settings:

- Border: *Color*: Red.
- Fill: *Color:* Yellow; *Style*: Solid.
- Font: *Bold* = True

d. Click OK (twice).

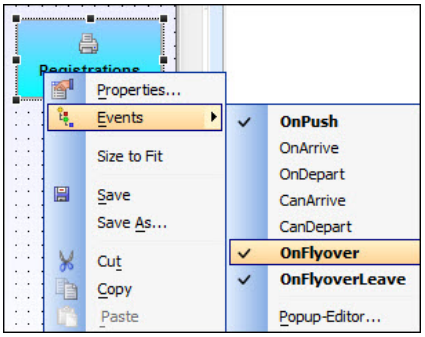
26. Save the form and go to Form View.
27. Hover over the button to see the new formatting.
28. Return to Design Mode.

Understanding button events

All objects, including the form itself, have events as well as properties. As expected, events are milestones in the life of an object. On your birthday, you might expect to mark the event with a cake. When you create a button, you want something to happen when it is pushed. This event is called **OnPush**.

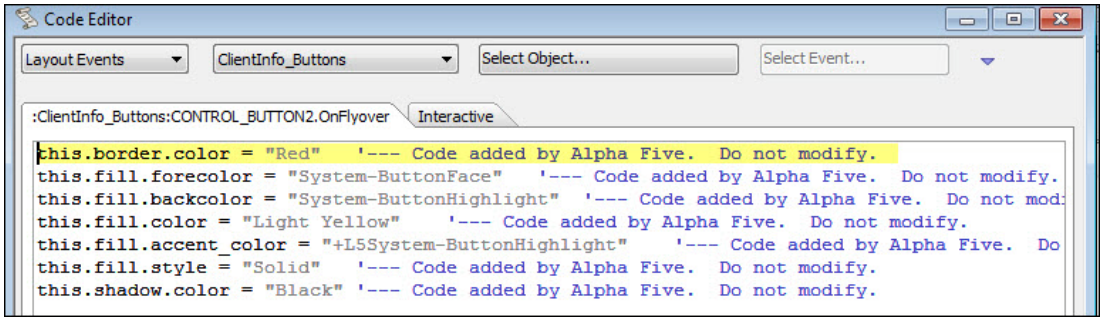
The list of events for a button are *OnPush*, *OnArrive*, *OnDepart*, *CanArrive*, *CanDepart*, *OnFlyover* and *OnFlyoverLeave*. Actions may be set for each.

- Alpha Five writes the Xbasic in the background as you make menu choices. Next, we'll take a look at the event for the OnFlyover effect that we just created.



29. Right click on the button you just made. Choose Events and inspect its submenu.

- *OnPush* is checked and bolded because the Button Genie placed our actions in that category when we created the button.
- *OnFlyover* is checked and bolded because we set Flyover in the Properties dialog box.
- *OnFlyoverLeave* is checked and bolded. The action to return to the original state is automatically created when *OnFlyover* is selected.
- Xbasic is constantly being written in the background as you make menu choices.

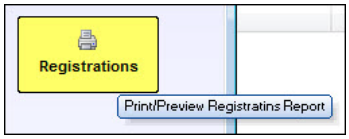


30. Choose **OnFlyover**.

- The Code Editor shows the Xbasic syntax.
- The properties are black.
- The settings are gray with quotes.
- The annotations — comments — are in blue.
- Notice that the annotations begin with an apostrophe [‘] telling Alpha Five to ignore the text.

MODIFY XBASIC

At the end of each line is the warning, “Code added by Alpha Five. Do not modify.” This notation simply means that things may not work properly if you modify the script. You can ignore the warning as we will do now – just be careful of the changes you make.



a. Change the border color (1st line) from red to black.

- Remember to keep the quotes!
- You may also copy any part of the code and put it in another script.

b. Save and close the Code Editor.

31. Press F11 to see the events for the following objects (double click works for some, but not all).
    - The background of the form, the tabbed object, the browse, a field.
32. Save the form and test the fly-over effects at Form View.
33. Close the form.

PLEASE TAKE A MOMENT TO READ THIS

Setting Multiple Actions for a Button

This is one of my favorite sections. It takes all those operations we created in Chapter 6 and ties them into a tidy package for the end user.

There will be two buttons. The first will chain several actions together. In addition to running the operations, we will open and close a report and finish with a browse that shows the state of the tables before deletion of the duplicate records.

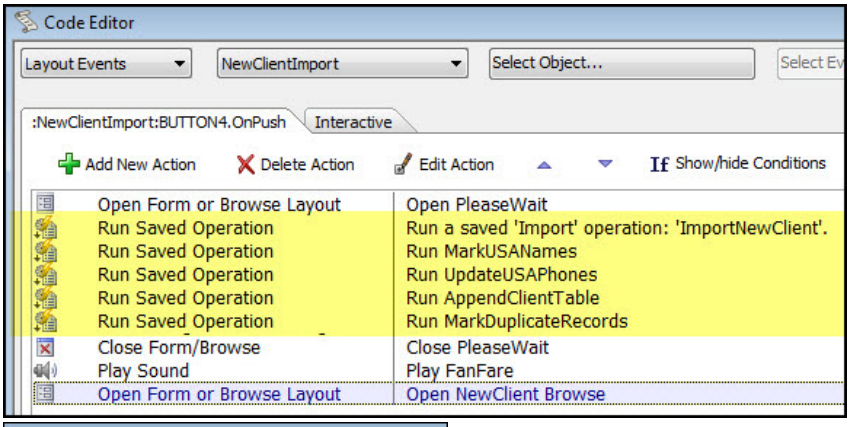
At that point, the end user can review the marked records and, if necessary, manually unmark any that should not be deleted. For example, we used First Name, Last Name and Company as the criteria for marking the records. We did not include Middle Initial. It is possible that John J. Jones and John K. Jones both work for XYZ Company. One of these records would need to be unmarked manually to exclude it from the delete operation.

The second button, already created for you, completes the script by deleting the duplicate records.

Action Scripting makes the process simple. Here's the plan.

a. When the button is clicked, a *PleaseWait* form will open. It asks the user to wait for the data processing to be completed.

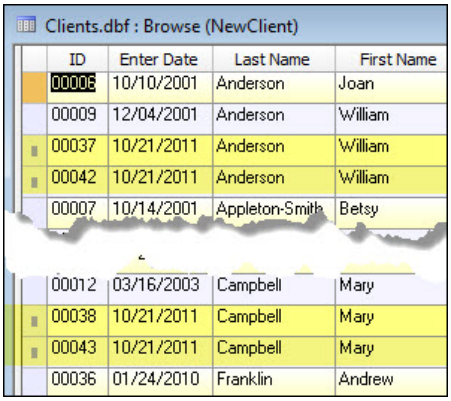
b. Then, five operations will run without interruption by confirmation dialogs.*



- Import records: **NewClientImport**.
- Mark records: **MarkUSANames**.
- Update records: **UpdateUSAPhones**.
- Append records: **AppendClientsTable**.
- Mark duplicate records: **MarkDuplicateRecords.**

c. The action continues by:

- Closing the *PleaseWait* form.
- Playing a sound to alert the user that the data processing is completed.
- Finally, a browse will open so the user may view the records to this point.



NOTE

We have included the above series of operations and will use them for the instructions. If you did the exercises in the previous chapter, you may use the ones you created.

Save, save, save as you go along.

1. Open the **NewClientImport_practice** form in design mode.

2. Create a new button under the Import List label.

*. If you completed the exercises in the previous chapter, you will be familiar with the confirmation dialog boxes that appeared as each operation was run. When running a series of operations, the dialog boxes would each need to be confirmed by the user in order to continue to the next operation. Since they are unnecessary for the first button, we will have the operations *Run Silently.* In other words, we will suppress the dialog boxes.

In the second button, we kept the dialog box that asks for confirmation of record deletion, in case the button was pressed accidentally.

3. *Label*: **Import Client List**. (Click Next)

ACTIONS #1-5

4. Click **Add New Action** and select the following choices.

   a. Category: **Operations**; Action: **Run Saved Operation**. (Click OK)

   b. *Tab: Operation: Type:* **Import records**; *Name*: **NewClientImport**.

SUPPRESS CONFIRMATION DIALOG

The next step suppresses the confirmation dialogs. If you forget it – and we do all the time, it can be added later.

c. *Tab: Options:* **Run silently**. (Click Next and Finish)

- As we discussed earlier, the Define Button genie does not have a way to enter comments nor does it have a Save button. So, we're off to the Script Editor.

5. Click **Launch Script Editor**.

   - The Code Editor opens and looky there – it created the comment for us. Comments for some actions are built in, but as we learned earlier, not for all.

- Next, we will add four more actions to the script. Begin each by clicking Add new action, click Next as necessary and end by clicking Finish.

#	Category	Action	Tab: Operation	Tab: Options	Name	Comment
2	Operations	Run Saved Operation	Delete, Mark/ Unmark records	Run silently =Yes	**MarkUSANames**	Run MarkUSANames
3	Operations	Run Saved Operation	Update records	Run silently =Yes	**UpdateUSAPhones**	Run Update USAPhones
4	Operations	Run Saved Operation	Append records	Run silently =Yes	**AppendClientsTable**	Run Append ClientsTable
5	Operations	Run Saved Operation	Delete/Mark/ Unmark record	Run silently =Yes	**MarkDuplicate Records**	Run Mark DuplicateRecords

- At this point, the list looks like the yellow section on page 304.

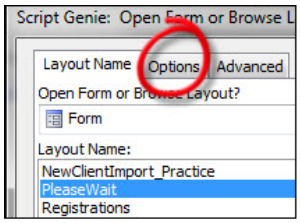
6. Click **Save** (button on Toolbar at top of screen).

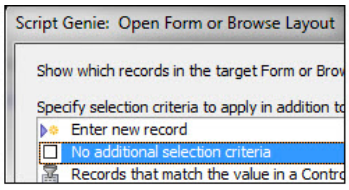
ACTIONS #5-6

Since this is a long script, we will add a *Please Wait* dialog. The results of the operations will be shown by a browse that opens last. As before, start each by clicking Add New Action, click Next as necessary and end by clicking Finish.

FORM / BROWSE OPTIONS

For the most part, adding forms and browses follows the same procedure as for Operations. There are some differences, however.





a. *Window position:* At the form selection screen, click the **Options** tab to see the additional settings for forms and browses. You can change the window type, the way it opens and/or its position.

- The next screen defines the way the form opens. It's not important here, since the Please Wait dialog doesn't have any fields. We'll detail it at "Password Protecting a Button" on page 313.

b. *Selection criteria:* Choose **No additional selection criteria.**

#	Category	Action	Layout Name: Browse/Form	Comment
6	Form/Browse	Open Form or Browse layout.	Browse: NewClientImport	Open NewClients Browse
7	Form/Browse	Open Form or Browse layout.	Form: PleaseWait	Open PleaseWait form

RE-ORDER ACTIONS

Now we need to adjust the order of the actions so the *PleaseWait* form opens first.

7. Use the **Move Action Up** arrow (next to Edit Action) to move the *PleaseWait* form to the top of the list.
   - Next, we will close the PleaseWait form.
8. Click on the last **Run Saved Operation** in the list.

ACTION #8

9. Click Add New Action.
   - **a.** *Category:* **Forms/Browses**; *Action:* **Close Form or Browse layout**. (Click OK)
   - **b.** *Close which Form or Browse window?* **Another Form or Browse window.**
   - **c.** How will you specify which Form or Browse to close? **By specifying the name of the Form or Browse *window*.**
   - **d.** Window Name: **Forms;** choose **PleaseWait**. (Click Next)
   - **e.** *Comment:* **Close PleaseWait form**. (Click Finish)

SOUND

Lastly, we will play a sound so the user knows the script is finished.

ACTION #9

10. Click **Add New Action**.
    - **a.** *Category:* **Miscellaneous**; *Action:* **Play sound**. (Click OK)
    - **b.** *Sound File* (.wav) to play: Navigate to the following:
    - C:\A5_MadeEasyDesktopBook\ABC_DesktopLessons\fanfare.wav\.
    - **c.** Click Open and then click Next.
    - **d.** *Comment*: Play Fanfare (or accept default). (Click Finish)
11. Move the sound to the 2nd to last position, just above the browse.
    - Refer to the screen shot on page 304 to be sure that the actions are in the correct order.

NOTE

Alpha Five is excellent about remembering to ask you to save your changes, but it's always best to do it yourself, too. Hate to lose work!

12. Click Save or press CTRL+S.
13. Close the Code Editor by clicking on the X in the upper right corner.

VERY IMPORTANT

Now we must save the form. We saved the script as we went along, but is necessary to save the form in order to keep the button and its actions.

14. Press CTRL + S to save the form.

- Time to test the button.

15. Go to Form View.

16. Click the button.
    - The script runs and ends by opening the browse (screen shot on page 304).
17. When the browse opens, press CTRL+HOME to go to the first record.
    - Notice that the duplicate records are marked.
18. Close the browse.

Editing the script

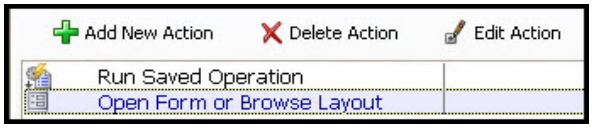
Did any dialog boxes open? If so, you can go back in and check the *Run Silently* box (see "Suppress confirmation dialog" on page 305).

- Design Mode > Select button > Right click > Events > On Push.
- Edit the action.
- Save the script, save the form and retest.

Completing the operations

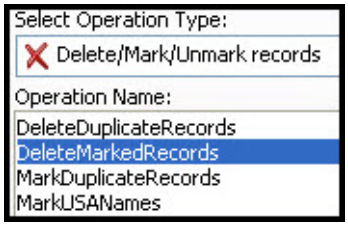
Deleting records is serious business. Now that we have had the opportunity to inspect the records and make any manual changes, we will use a second button to delete the marked records. We have created that button for you, but let's take a look at its events before using it.

19. Return to Design Mode.
20. *Delete Marked Records* button: Choose **Events > OnPush**.



- Notice that the Actions have not been commented, so it is hard to know which Operations will run. Let's fix that now.

21. Select *Run Saved Operation;* then click **Edit Action**.



- *Operations* tab: Operation type = Delete/Mark/Unmark Operation; Operation Name =DeleteMarkedRecords.

22. Click Next twice and enter the Comment: **Run DeleteMarkedRecords**. (Click Finish)
    - The comment appears and, in the future, we won't have to guess!
23. Repeat for the browse.

IMPORTANT NOTE

This script **does not run silently** because it is possible that the button could be clicked accidentally.

24. Save the changes and test the button.
25. Close the form or keep it open if you are continuing to the next section.

SCRIPT PROGRESS

To show the user the progress of a very long script, see E*nhancing Your Desktop App!* by Susan Hussey Bush. Available at www.libertymanuals.com.

Commenting

As you noted in the above exercise, commenting is invaluable. Without it, you may not be able to ascertain the original intention of the action. As Dr. Peter Wayne, Alpha Five Xbasic expert, says, "There are two types of developers: Those who comment and those who wish they had!"

As we have seen, there are two places to enter comments:

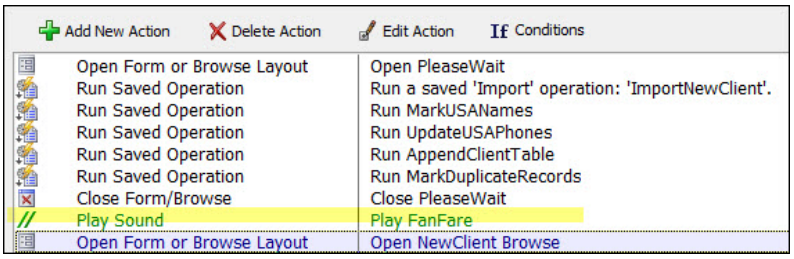
- In the final dialog box of the Action Scripting genie.
- The tiny pencil in the lower right corner of the Code Editor.

- Next, we will look at the second use of the word **Comment**.

Temporarily stopping an action

Sometimes when you are designing a script, you need to stop an action while testing and then you want to add it back in later. You can stop any element of an Action Script at any time. It will remain in the list, but the action will not run with the others in the script. This is called **Commenting Out**. If you decide to add it back in, that is called **Commenting In.** Both actions are available from the Code Editor right click menu.

- This exercise continues from the previous one. NewClientImport_Practice should be open in Design Mode.
- If you did not do the previous exercise, open NewClientImport in Design Mode



1. Import List button: Right click > Events > OnPush.

2. Right click on **Play Sound** and choose: **Comment In/Out this Action**.

- The action and its description turn green and the icon at the left turns to **//.**

3. To test, save your changes, go to Form View and test the button. The script runs, but the sound does not play.
4. Go back in and **Comment In** the **Play Sound** Action. (Click **Save**)
5. Click the *Delete Marked Records* button to clear the duplicates creating during the above testing.
6. Save and close the form.

PLEASE TAKE A MOMENT TO READ THIS

Understanding Selection Criteria for Buttons

"Selection criteria" is a type of filter that defines the way forms, browses and reports open when a button is clicked. It is in addition to any filtering done at the layout level. We will demonstrate with two different types of buttons.

- **Form buttons.** These are the buttons that we've been using so far in this chapter.