

# The Hidden Treasure of Action Scripting

Presented by Frances Peake

## ***Introduction***

Many Alpha Five users overlook Action Scripting and the Xdialog Genie, assuming that they are not powerful enough or are too restrictive. Discover how you can save time and build your Xbasic and Xdialog skills by letting the Action Script genies write your code. Then with a few simple steps, modify and enhance it to meet complex requirements.

## ***How I became an Xbasic programmer***

I never considered myself to be a programmer, yet I have built a 15-year career developing database applications with Alpha Five. Business people are attracted to Alpha Five because it was designed for them, not for programmers. With Alpha Five, you really *can* solve most database challenges without writing code.

I respect and admire those in the Alpha developer community who, when faced with a database challenge, reach for Xbasic first. They are adept at Xbasic programming. They prefer it and often recommend it over using Action Scripting, Genies, Saved Operations and Web Components. Ironically, these powerful-but-easy-to-use tools are exactly what make Alpha Five so much more appealing than its competitors!

As a non-programmer, I reach for Action Scripting and the other “non-programmer” tools first. Nevertheless in recent years I have met with some challenges that required me to learn more and more Xbasic. I have come to appreciate the higher degree of control and efficiency that it can provide.

These days I create a lot of hybrid scripts – part Action Scripting and part Xbasic. I often start in Action Scripting and then convert some of the actions to Xbasic for further customization. (The same can be accomplished with Operations by defining them in the Control Panel and converting them to Xbasic.) Studying the code that Action Scripting generates is a great way to learn Xbasic!

## ***What you will learn from this presentation***

- Highlights from the Xdialog Action Script genie
- When to convert Action Scripts to Xbasic
- How to integrate Action Scripting and Xbasic

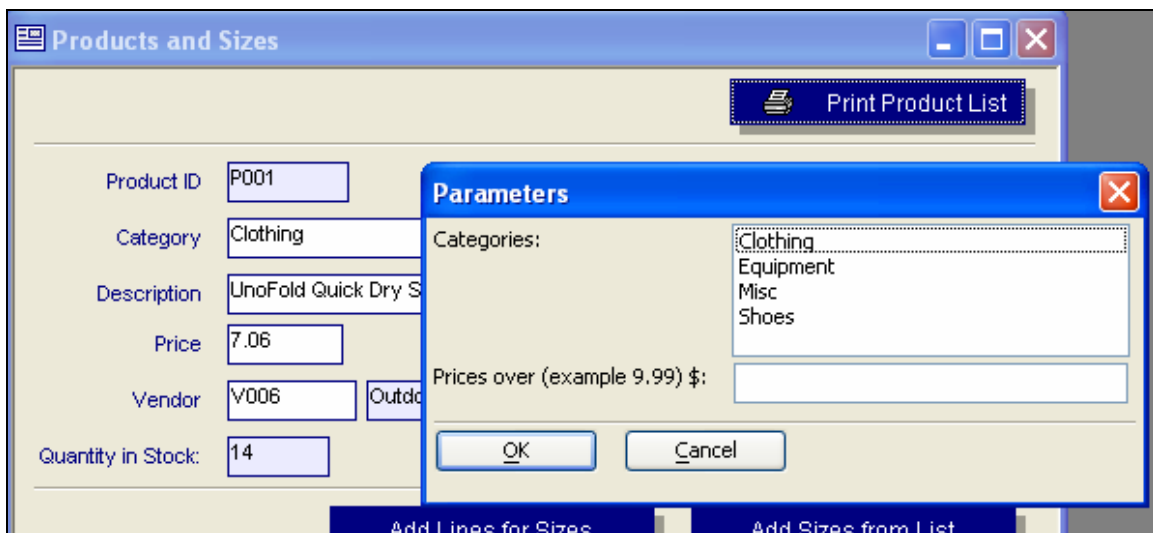
The following examples are contained in the A5ConfActionScripting database, which can be downloaded from the conference server during the conference, or from

## The Hidden Treasure of Action Scripting (cont.)

<http://www.proctorandpeake.com/a5conf/A5Conf2006ActionScripting.zip>. A copy of this document in PDF format is included in the zip file.

### Example #1 Action Scripting: “Print Product List” Button

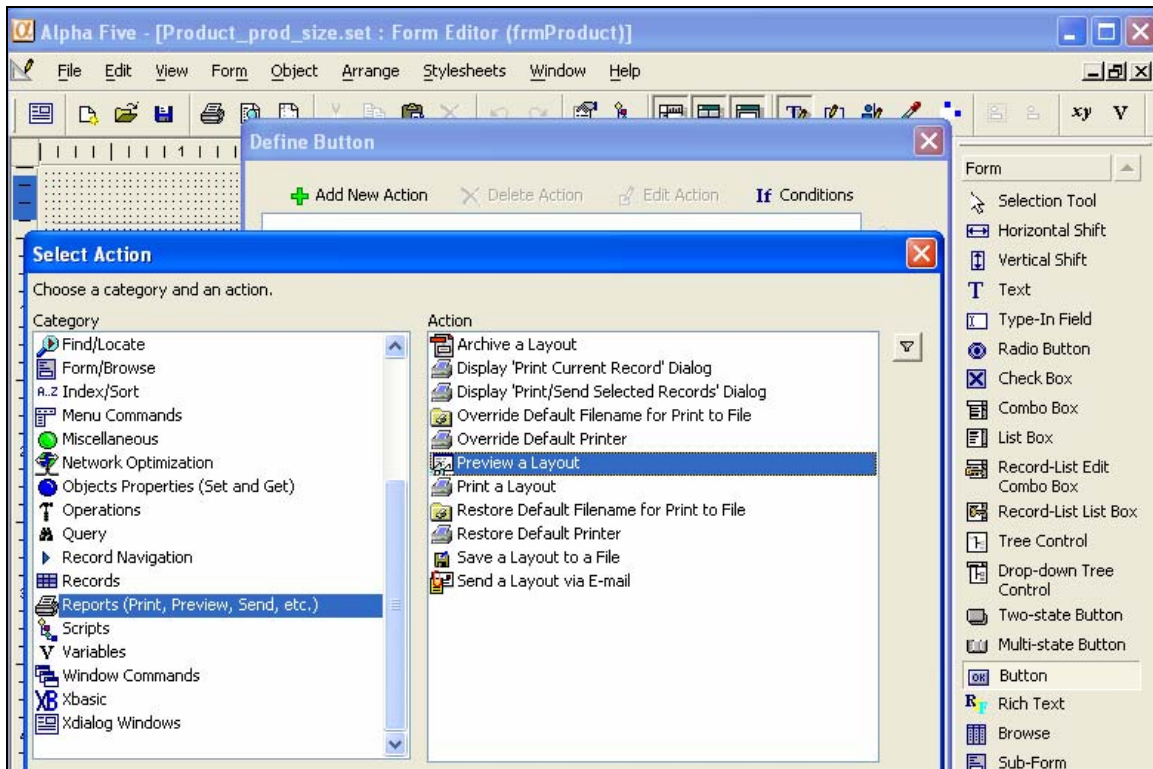
The Print Product List button illustrates a very common use of Action Scripting. Rarely do you want a report to print every record every time. The actions for previewing, printing, e-mailing and archiving layouts give you an easy way to select which records will print.



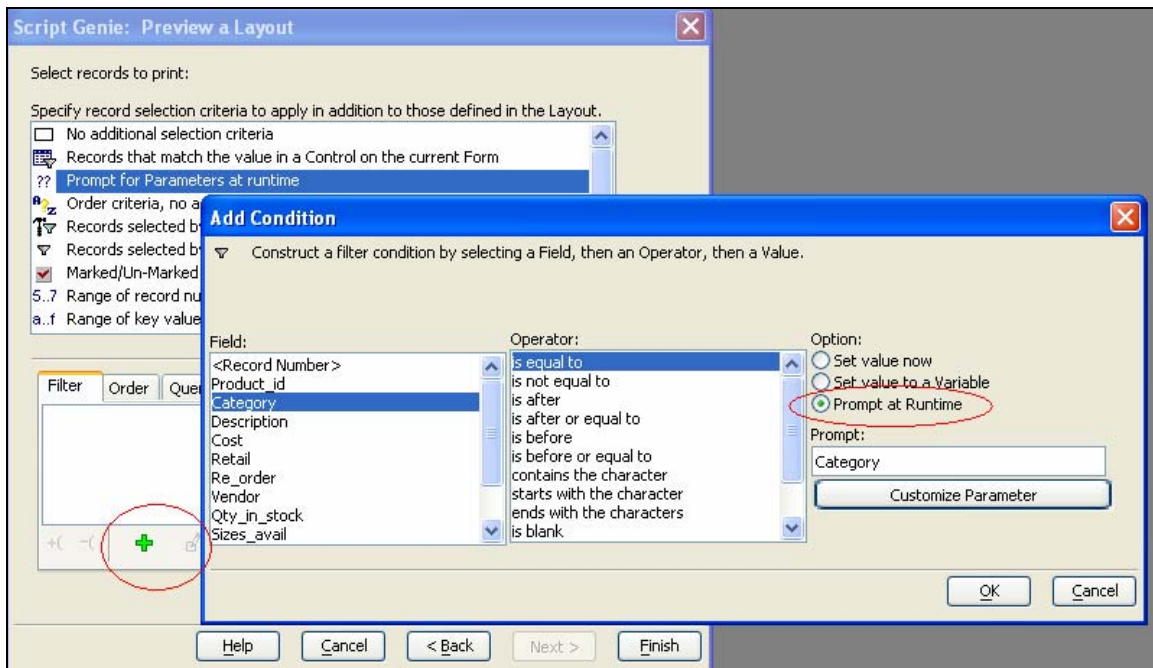
Here are the steps to create the Print Product List button’s action script.

1. Edit the form frmProduct. In the Form Editor choose the Button tool on from the Toolbox. With your mouse, draw a button. This will start the Define Button genie.
2. In the Define Button genie, choose Use Action Scripting and click Next.
3. Click Add New Action to go to Select Action. Choose the Reports category and the Preview a Layout action, as shown in the illustration below.

# The Hidden Treasure of Action Scripting (cont.)

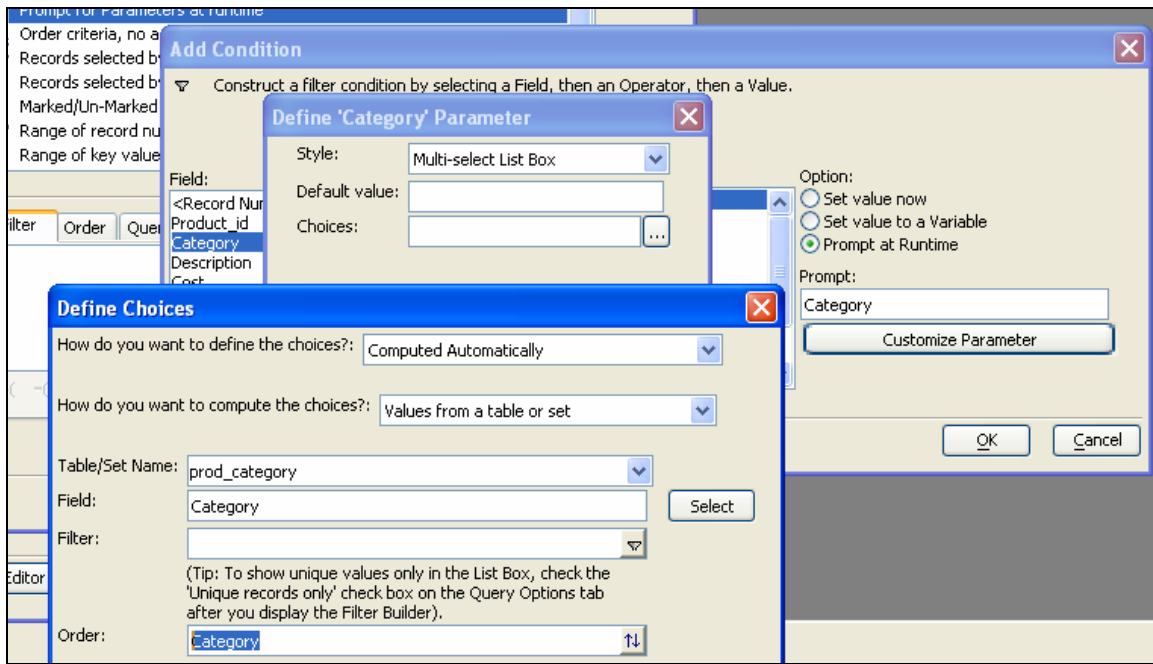


4. Click OK. On the Preview a Layout genie, choose rptProduct. Click Next to select the records to print.



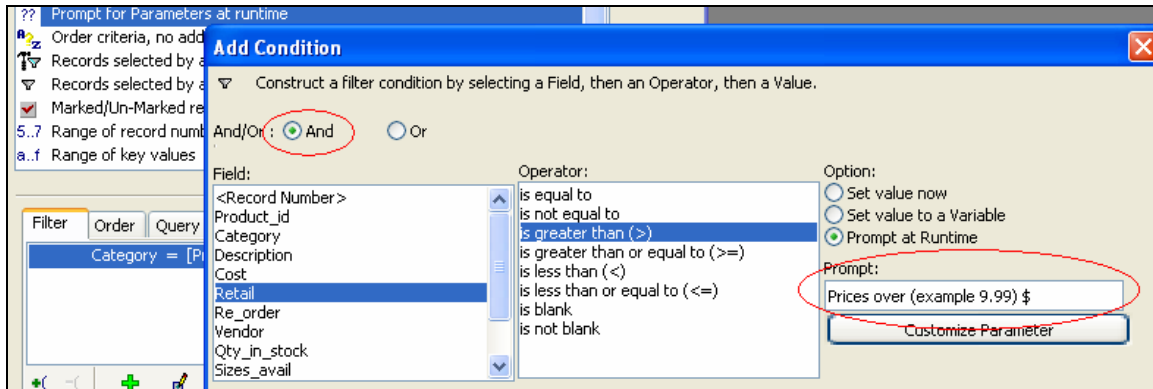
## The Hidden Treasure of Action Scripting (cont.)

5. Choose Prompt for Parameters at Runtime, then click the green plus sign to add a condition, as shown on the previous page.
6. Choose Prompt at Runtime and click the Customize Parameter Button.
7. Under Define 'Category' Parameter choose Multi-Select List Box.
8. To the question "How do you want to define the choices", select Computed Automatically. Then fill in the dialog box as shown below.



9. Click OK 3 times to return to the Preview a layout dialog.
10. Now you are ready to enter the next condition, "Prices over \$". Click the green plus sign and fill in the Add Condition dialog box as shown below.

## The Hidden Treasure of Action Scripting (cont.)



11. Click OK to return to the Preview a layout dialog, then click Finish twice to return to the form editor.

12. Save the form and test the button you created.

### Example #2 Xdialog Genie: “Add Item (2 dialogs – conditional actions)” Button

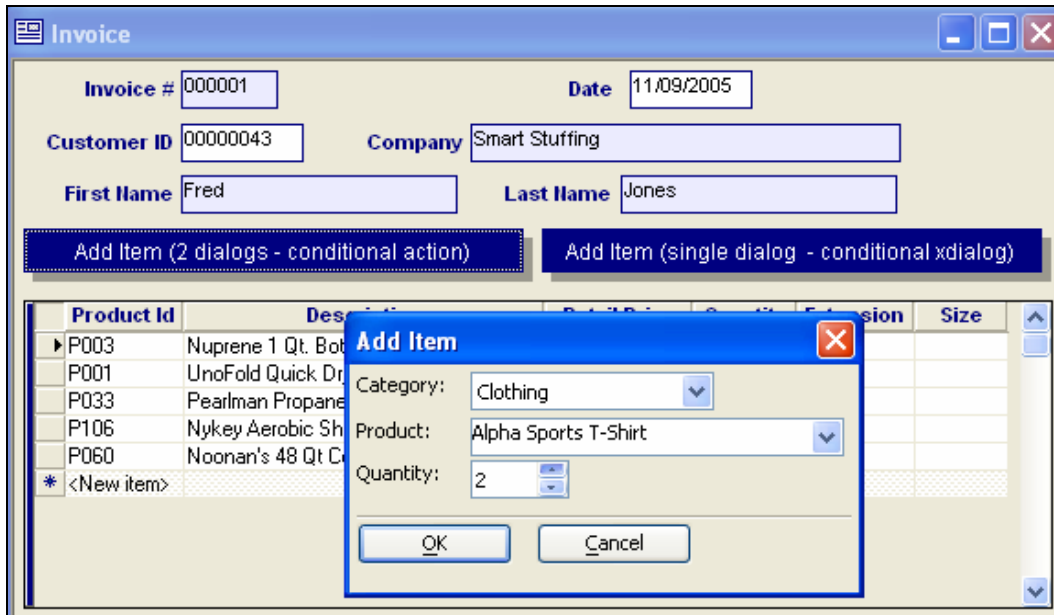
Some products in the sample database, such as clothing, are available in different sizes. This is indicated by the Sizes\_avail field in the Product table. The sizes for these products are contained in the Product\_sizes table.

The button in this example adds an item to the invoice on the Invoice form. The user can choose a product to add to the invoice and if the product is available in sizes, then a second prompt lets them choose the size.

This example shows:

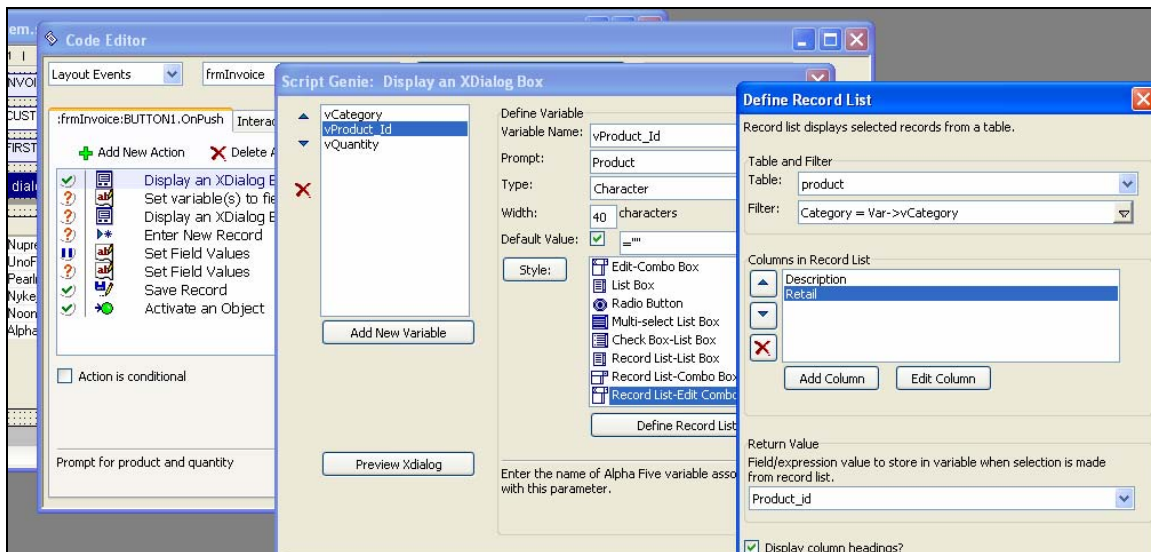
- Two different styles of combo box
- Getting values from a table into a variable
- Using conditions in Action Scripting

# The Hidden Treasure of Action Scripting (cont.)



Rather than recreating this button, you will go into the Action Script editor to study how it is put together.

1. Edit the invoice form frmInvoice. In the Form Editor, right-click the Add Item (2 dialogs – conditional actions)” button. Choose the OnPush event.
2. The Code Editor will open, showing you the Action Scripting actions. Use the Edit Action button to step through each genie. When finished, click Finish.

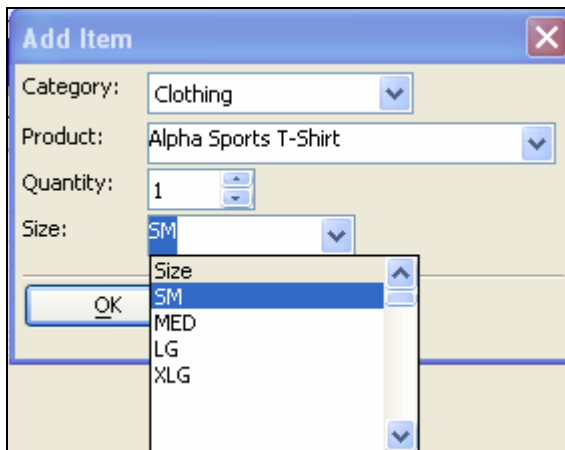


## The Hidden Treasure of Action Scripting (cont.)

### Example #3 Xdialog Converted and Modified: “Add Item (single dialog – conditional xdialog)” Button

This button accomplishes the same thing as the button in #4, but the size choices appear on the same dialog box as the choice of product. It uses an Xdialog condition to offer this choice only when sizes are available for the chosen product.

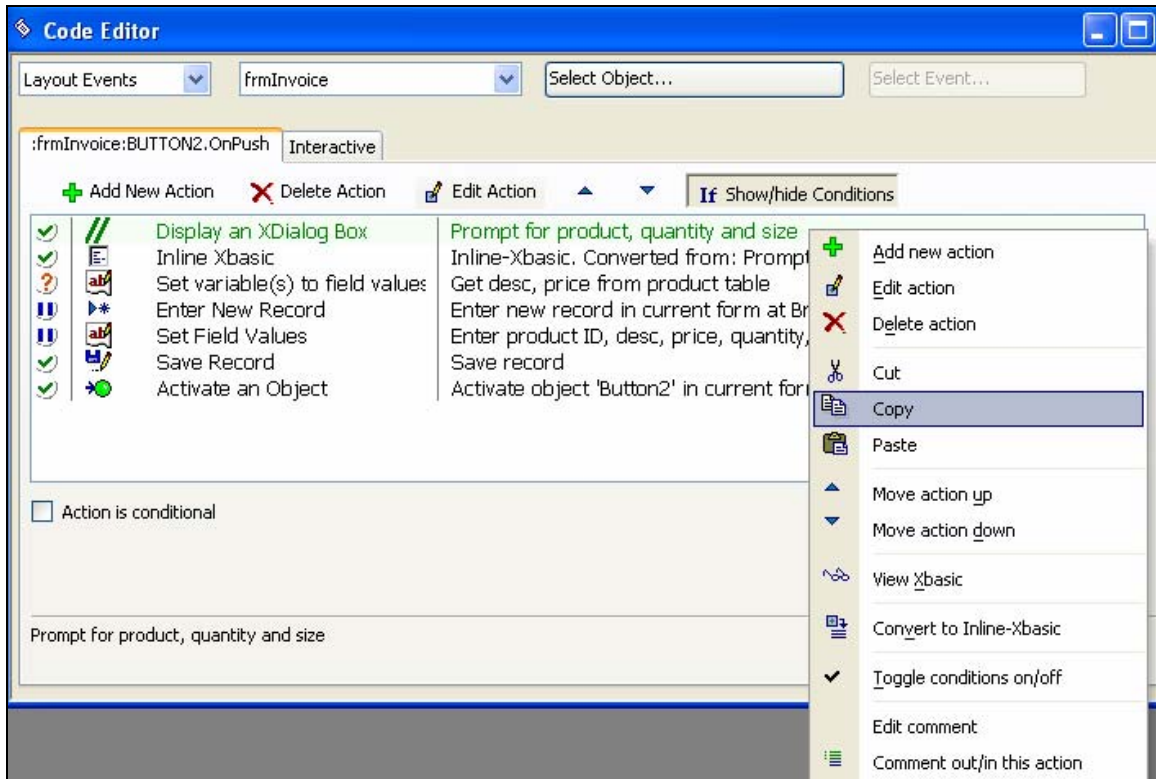
This example shows how you can convert the Show an Xdialog action to Xbasic and modify the Xdialog code to add a condition. In addition to adding the condition, we will use a trick to force the sort order of the sizes combo box column. We want the sizes to sort on the Sort\_by field rather than sorting alphabetically by the letter for size.



Here are the steps to convert the Xdialog and make the size prompt conditional:

1. Start by editing the Add Item (single dialog – conditional xdialog)” button’s OnPush event.
2. Edit the commented-out Show an Xdialog action so that you are familiar with its variables and controls. This is the action that we will copy and convert to Xbasic.
3. Copy the Show an Xdialog action by right-clicking on it and choosing Copy as shown below. When prompted, choose to copy the Current Action.

## The Hidden Treasure of Action Scripting (cont.)



4. Right-click again and choose Paste.
5. The action is commented out, which is indicated by the green // to its left. Right-click and choose Comment out/in this action to uncomment it.
6. Right-click again and choose Convert to Inline-Xbasic.
7. Click Edit Action. If you do not see line numbers to the left of each line then right-click and choose Line Numbers. You should see this:

## The Hidden Treasure of Action Scripting (cont.)

```

1 'Prompt for product, quantity and size
2 DIM SHARED vCategory as C
3 DIM SHARED vProduct_Id as C
4 DIM SHARED vQuantity as N
5 DIM SHARED vSize as C
6 DIM SHARED varC_result as C
7 vProduct_Id = ""
8 vQuantity = 1
9 vSize = ""
10 auto_list_vCategory = table.external_record_
11
12 temp_count = w_count(auto_list_vCategory,crl:
13 DELETE a_vCategory
14 DIM a_vCategory[temp_count] as c
15 a_vCategory.initialize(auto_list_vCategory)
16 ok button label = "<OK"

```

Line: 1 of 60 | Column: 1 | <actions<[1]<display=""> <sl | NUM

Zoom to Resizable Window

8. Click Zoom to Resizable Window.

9. To make the sizes combo box conditional, add conditions at lines 32 and 35, as shown in bold below:

```

{region}
Category:| [%v%.22vCategory^=a_vCategory!vCategory_changed];
Product:| [%@vProduct_Id_rl_def%.40vProduct_Id!vProduct_Id_changed];
Quantity:| [%z;s1,100%.10vQuantity!vQuantity_changed];
{condition=(vSizes_Avail = .T.)}
Size:| [%@vSize_rl_def%.20vSize];
{condition=.T.}
{endregion};

```

10. At line 56, add a lookup expression to update the value of the vSizes\_Avail variable whenever the Product ID changes, that is, when the user chooses the product from the combo box. Note that we are using:

```

If a_dlg_button = "vProduct_Id_changed" then
  vSizes_Avail = lookup("F",vProduct_Id,"Sizes_avail","product","Product_Id")
  vSize_rl_def = replace_parameters(vSize_rl_def_orig,local_variables())
  a_dlg_button = ""
end if

```

## The Hidden Treasure of Action Scripting (cont.)

11. Now, for the trick to sort the sizes by the Sort\_by field instead of the Size field, change line 23. Where “25” is circled below, change that to 00.

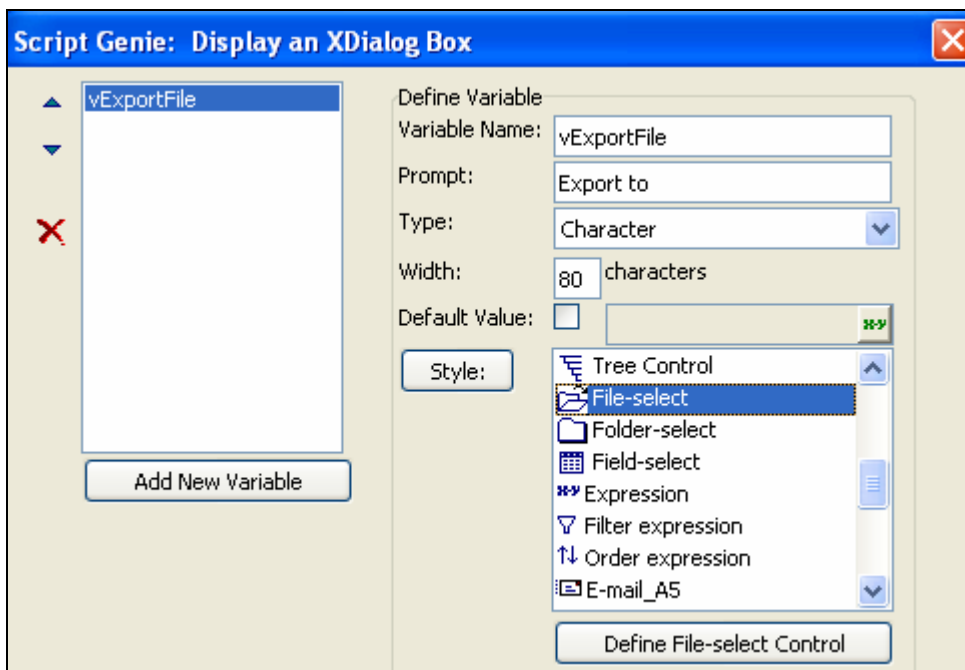
```
23 Dim vSize_rl_def_orig as C
24 vSize_rl_def_orig = "ke=prod_size,{keylist_build(\"H=.025,1:25[Sort_By],2:25[Size]\",
25 Dim vSize_rl_def as C
```

### Example #4 Export with Prompt for Filename: “scpProductExport” Global Script

You can define an Operation in the Alpha Five Control Panel and then convert it to Xbasic. This example uses an Xdialog to prompt the user for the export file location and name. That value is then plugged into the converted Export operation.

Here are the steps:

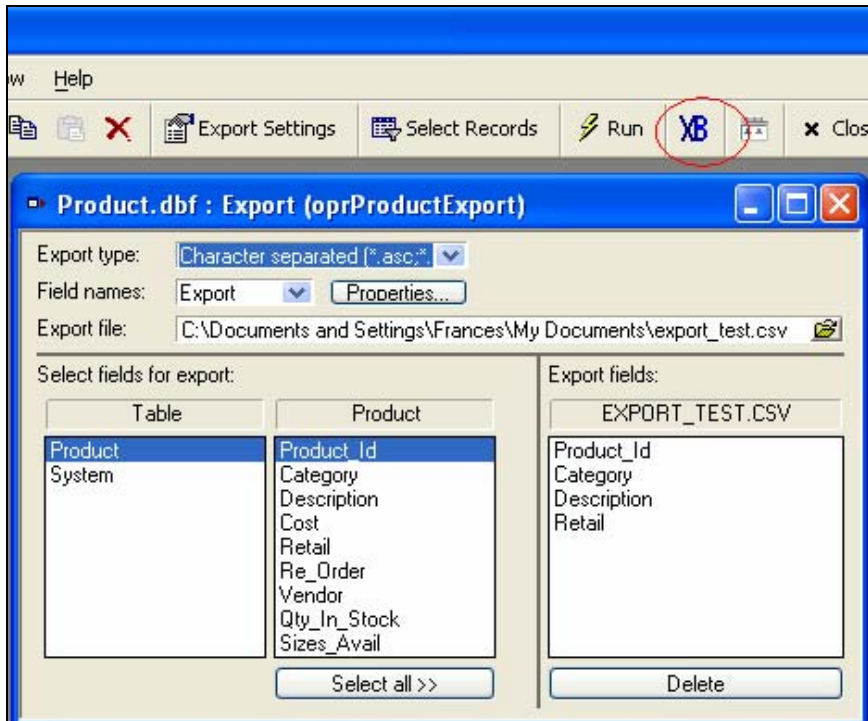
1. Begin on the Code tab of the Alpha Five Control Panel. Create a new script using Action Scripting.
2. Choose the Display an Xdialog action. Create a variable named vExportFile and choose the File-Select control as shown below. You may also change the file-naming pattern by clicking the Define File-select Control button.



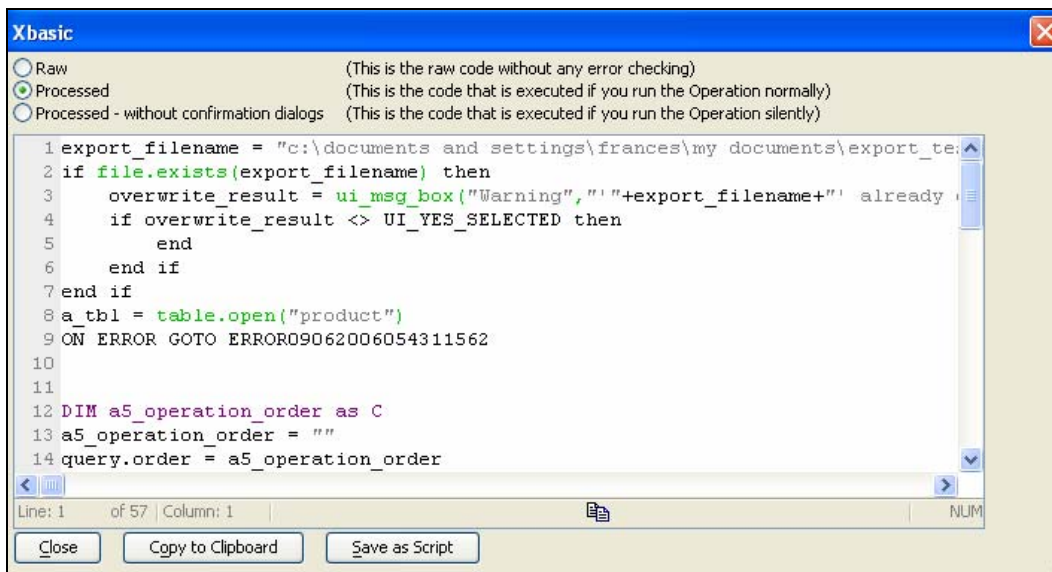
3. Click Next and give the dialog a title, “Export”. Click Finish.

## The Hidden Treasure of Action Scripting (cont.)

- Minimize the Code Editor and go to the Operations tab of the Control Panel. Edit (Design) the Export operation named oprProductExport. Click the Show Xbasic Code button on the toolbar, which is circled in red in the screenshot below.



- Choose the Processed radio button then click the Copy to Clipboard button.

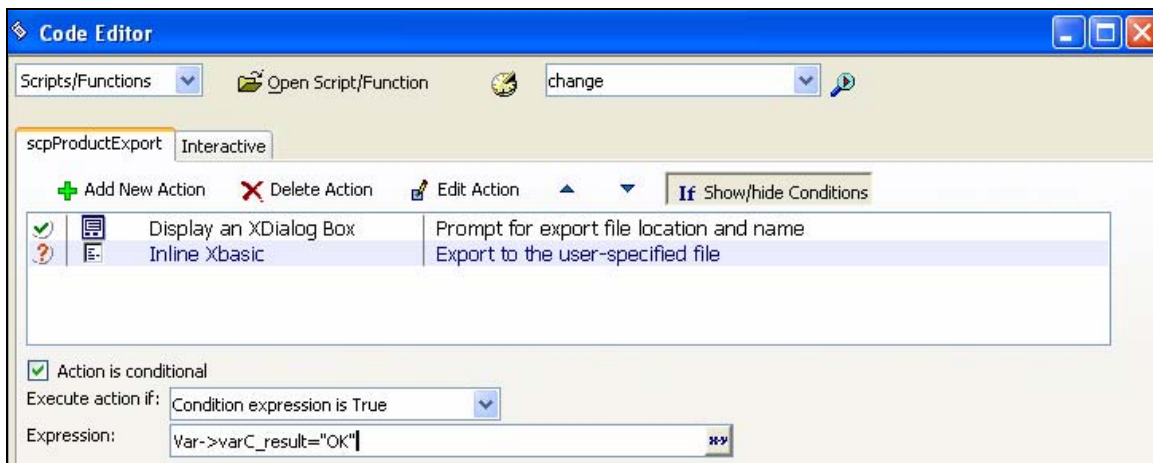


## The Hidden Treasure of Action Scripting (cont.)

6. Close the Operations window and return to the Code Editor. Add a new action, Inline Xbasic.
7. Click Zoom to Resizable Window. Right-click in window and choose Paste to paste the converted Export code. Change line 1 to use the variable vExportFile which will contain the file name and location that the user chose on the File-select dialog box:

```
export_filename = vExportFile
```

8. Make the action conditional with the expression shown below (the Expression Builder will help you choose the varC\_result variable). This will assure that the export runs only if the user clicked OK on the File-select dialog box.



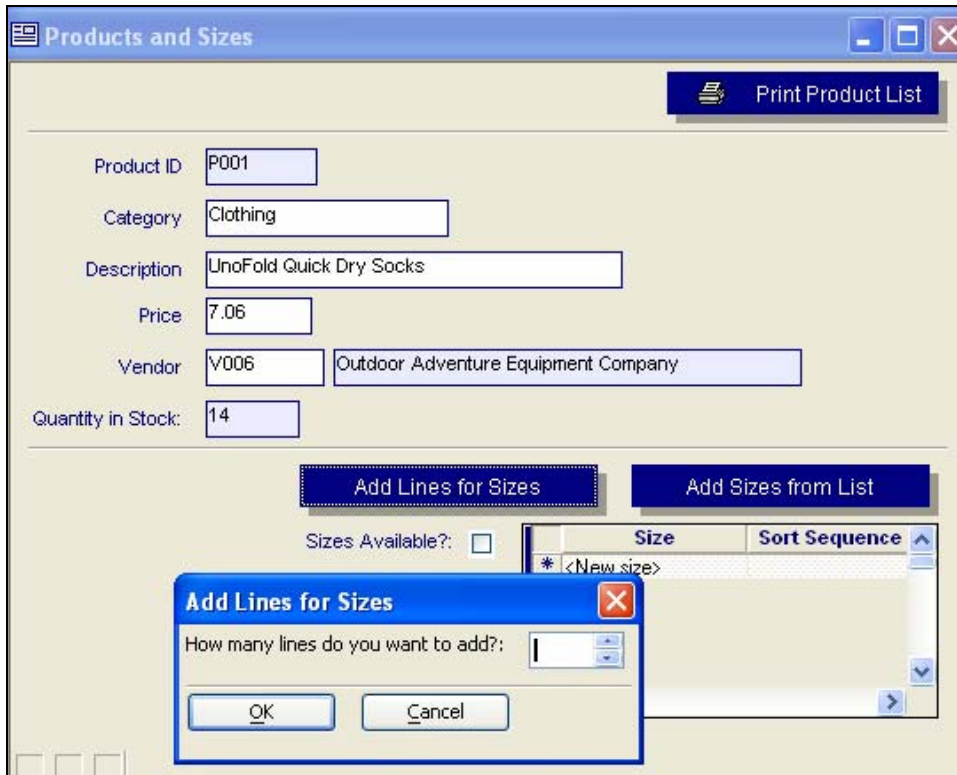
9. Save the script.

### Example #5 Adding records/FOR...NEXT loop: “Add Lines for Sizes” Button

Action Scripting does not do FOR...NEXT or WHILE...END WHILE loops. These must be defined in Xbasic. However, you can use the Action Script Genies to generate the code that will be used within the loop.

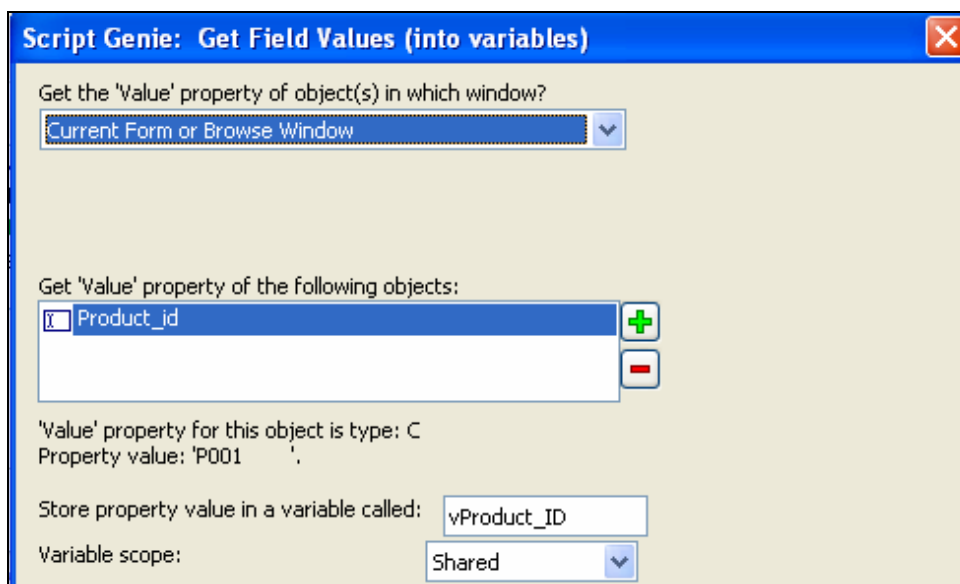
The “Add Lines for Sizes” button on the Products and Sizes form helps the user enter the sizes for a new product quickly. It assigns the sequence numbers (e.,g. 001, 002, etc.) that will be used elsewhere to sort the sizes logically instead of alphabetically.

# The Hidden Treasure of Action Scripting (cont.)



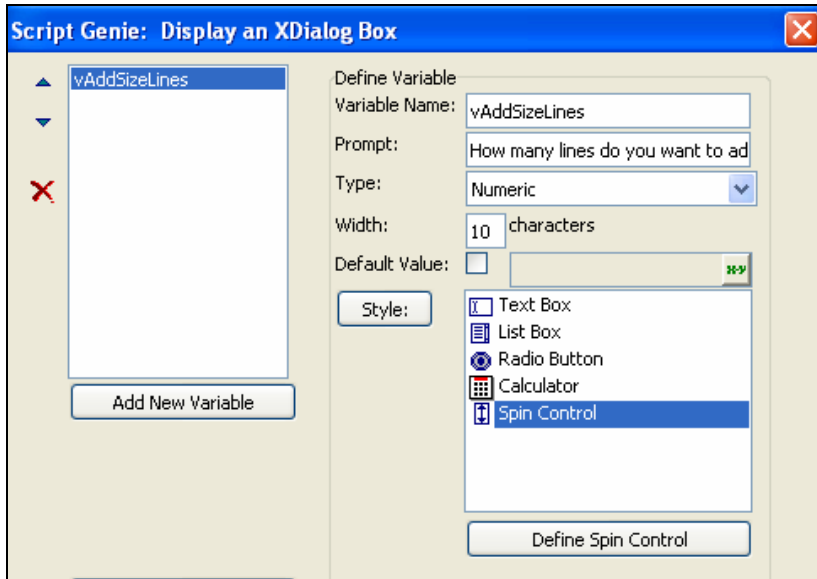
The Add Lines for Sizes button was created using the techniques covered in previous examples. Here are the key actions that make this button work.

1. The product ID for the record on the screen is saved to the variable vProduct\_ID.

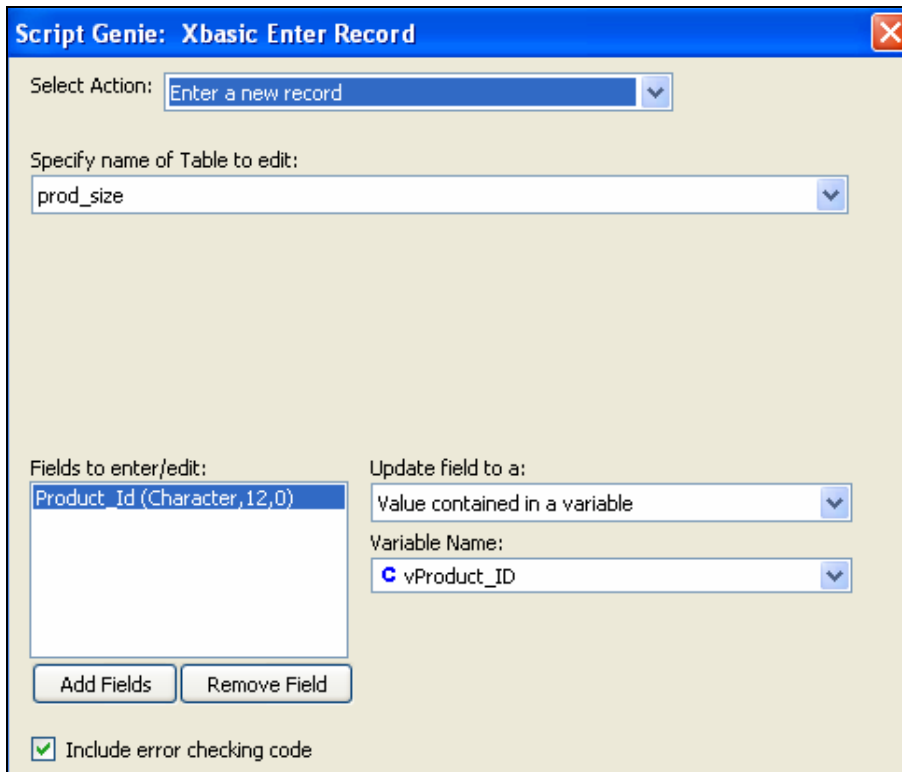


## The Hidden Treasure of Action Scripting (cont.)

- The user is prompted to enter the number of lines to add. This is stored in the vAddSizeLines variable.



- The Xbasic Enter Record action adds a new record to the Prod\_Size table.



## The Hidden Treasure of Action Scripting (cont.)

- The Xbasic Enter Record action is converted to inline Xbasic. The lines shown in bold are added to the script. The FOR...NEXT loop adds the number of size records that the user requested.

```
'Open the table that you want to update...
dim tbl as p
tbl = table.open("prod_size",FILE_RW_SHARED)
```

```
dim i as n
FOR i = 1 TO vAddSizeLines
```

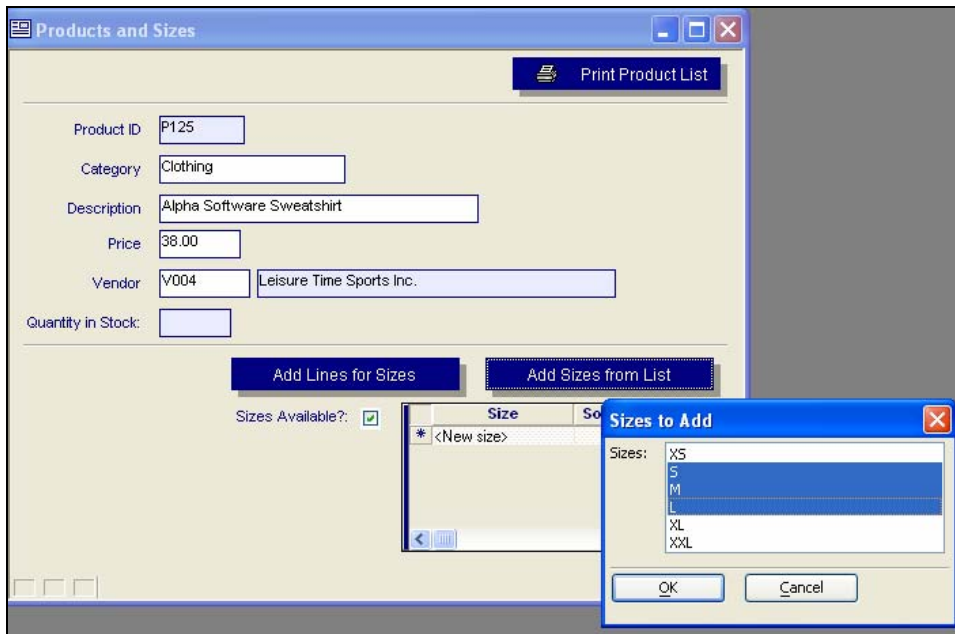
```
tbl.enter_begin()
tbl.PRODUCT_ID = convert_type(vProduct_ID,"Character")
tbl.Sort_by = padl(ltrim(str(i)),3,"0")
tbl.enter_end(.t.)
```

```
NEXT
```

```
tbl.close()
```

### Example #6 Adding Records with Listbox Values: “Add Sizes from List” Button

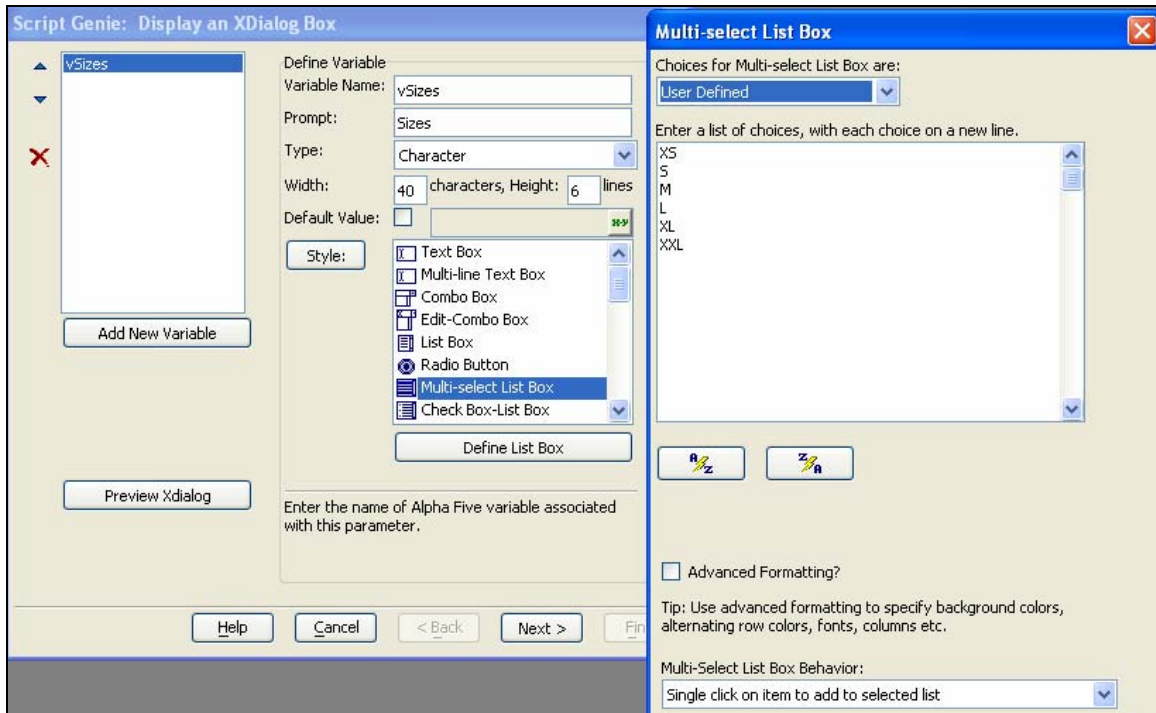
This example takes the previous button one step further. It allows the user to choose which sizes they want add. This is accomplished with a multi-select list box.



## The Hidden Treasure of Action Scripting (cont.)

Key steps are:

1. Create an Xdialog Box with the Multi-Select List Box control. The user's choices are saved in the vSizes variable.



2. Convert the Xbasic Enter Record action to Xbasic and save it in as a Global Script. This allows you to use the full Xbasic Code Editor to edit the script. This script, shown on the next page, was saved with the name scpProductAddSizes and appears on the Code tab of the Control Panel.

## The Hidden Treasure of Action Scripting (cont.)

```
dim tbl as p
tbl = table.open("prod_size",FILE_RW_SHARED)

dim count as n
count = w_count(vSizes,crlf())

dim i as n
FOR i = 1 TO count
    dim vSizeline as c
    vSizeLine = word(vSizes,i,crlf())

    tbl.enter_begin()
    tbl.PRODUCT_ID = convert_type(vProduct_ID,"Character")
    tbl.Sort_by = padl(ltrim(str(i)),3,"0")
    tbl.Size = vSizeline
    tbl.enter_end(.t.)
NEXT

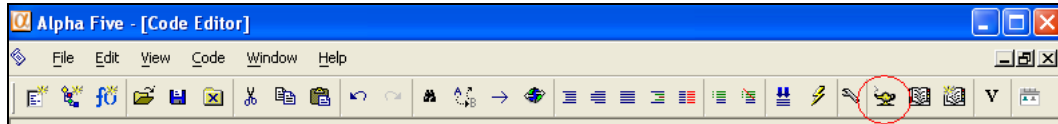
tbl.close()
```

3. Use the Run a Script action to run the saved script.

# The Hidden Treasure of Action Scripting (cont.)

## Tips

- You can use the Xbasic Script Genie in the Xbasic Editor to generate code for you! It will open the Action Script Editor, let you choose and customize an action, and then give you the option to paste the code into the Xbasic Editor.



- You can attach a script (Action Script or Xbasic) to a form or field event such as OnChange, but these events are difficult to test. It is easiest to start by putting your script on a button or a global script. Test it there and then paste the actions or code to the event.
- Before you convert an Action Scripting action to Xbasic, copy the action and comment it out. This prevents it from running. The unconverted action will serve as a backup in case you have trouble with the changes you make in Xbasic. Later on, it will help understand the Xbasic.

+ Add New Action		X Delete Action		Edit Action		If Show/Hide Conditions	
✓	//	Display an XDialog Box		Prompt for product, quantity and size			
✓	E	Inline Xbasic		Inline-Xbasic. Converted from: Prompt for product, quantity and size			
?	ab	Set variable(s) to field values		Get desc_price from product table			

Note: Right-click on the action to choose Comment Out/In This Action.

- Operations that you define in the Control Panel can easily be converted to Xbasic. Edit the operation and use the toolbar button shown below.

