Getting started with Xbasic Programming:
A practical guide

Presented by Frances Peake of Proctor & Peake, Inc.

ATEC2008 Conference Session Description

To the non-programmer, getting started with Alpha Five’s Xbasic programming language can be intimidating. But some tasks just cannot be accomplished without Xbasic. This session will focus on the most useful Xbasic skills and show you techniques that make it easier to learn, test and use Xbasic in your Alpha Five applications. Topics will include: variables, reading from and writing to tables, conditional code and looping, working with text strings, creating user-defined functions and best practices.

Xbasic - What is it and what is it good for?........................................................................... 1
  What is Xbasic? .................................................................................................................. 1
  What is Xbasic good for?................................................................................................. 2
  Alpha Five has genies and Action Scripting. Why use Xbasic?........................................ 2
  Terms you’ll need to know .............................................................................................. 3
Xbasic in Action ................................................................................................................. 3
  Reading From and Writing To tables............................................................................... 4
  Variables .......................................................................................................................... 5
  Using Action Scripting to Generate Xbasic Code ............................................................ 7
  Conditional Statements ................................................................................................. 10
  Text String Manipulation ............................................................................................... 16
  Looping ............................................................................................................................ 18
  Converting Saved Operations to Xbasic ....................................................................... 21
  Web Applications Genies ............................................................................................. 24
  Xbasic Tips ...................................................................................................................... 30
More to learn on your own ............................................................................................... 31

**Xbasic - What is it and what is it good for?**

**What is Xbasic?**

Xbasic is Alpha Five’s built-in programming language. It is similar to Visual Basic but it contains many specialized and high-level commands for use in Alpha Five.

The Xbasic Code Editor is available in every edition of Alpha Five except the Application Server. Alpha Five or the Application Server must be running to run Xbasic code.
What is Xbasic good for?

- Automating a series of tasks.
- Prompting users for input and giving them feedback.
- Reading from and writing to Alpha Five and other databases and files.

Alpha Five has genies and Action Scripting. Why use Xbasic?

Xbasic can be a more efficient way to manage repetitive or similar steps. Xbasic scripts, when thoughtfully written, will run faster.

You cannot use Action Scripting in web components and pages. You may only use Xbasic. Nevertheless Action Scripting can be used to generate code that you can then paste into your web application.

Alpha Five’s Operations let you import, export and manipulate your data in many different ways. Saved Operations have limitations. For example, in an Export Operation location and file name for the export file cannot be changed at run time. You may want to prompt the user for the export file name. To accomplish this, you must convert your saved operation to Xbasic.

The sections that follow provide details and examples to illustrate why Xbasic is worth learning.
Terms you’ll need to know

These definitions are given in the context of Alpha Five. You will see them in action later in this session.

**Script** – A list of commands that run automatically. In Alpha Five, you use the Xbasic programming language to write scripts.

**Function** – A type of script that can accept values as input and return values as output. Alpha Five comes with many built-in functions. You can also write your own functions with Xbasic.

**Variable** – A named value that can be changed. Variables are used in Alpha Five programming to hold temporary values and pass the values to scripts and functions.

**Expression** – A combination of values, operators and functions that result in a value. For example, the expression QUANTITY * UNIT_PRICE will result in the extended price.

**Flow Control Statement** – A conditional or looping structure that controls the order in which commands will be executed in a script. Alpha Five’s Xbasic contains many types of statements, including IF…THEN…ELSE…ENDIF and WHILE…END WHILE.

**Event** – An occurrence, such as a key press, form open or placement of the cursor. In Alpha Five you can attach scripts to events.

**Xbasic in Action**

This section will use practical examples to teach fundamental Xbasic skills:

- Reading From and Writing to Tables
- Variables
- Using Action Scripting to Generate Xbasic Code
- Conditional Statements
- Text String Manipulation
- Looping Statements
- Converting Saved Operations to Xbasic
- Using Web Applications Genies

Preparation: The exercises use Alpha Five Version 9 and the sample database ATEC2008_Xbasic. Copy/extract this database into a new folder and open it.
Reading From and Writing To tables

One of the most common reasons to use Xbasic is to interact directly with data in tables. That is, to go behind the user interface (e.g. desktop forms or web grid components) and read through records, add records or update records.

EXERCISE 1

In this exercise you will use the Interactive Window to add a record to the Product table. The Interactive Window lets you work “live” by typing in Xbasic commands.

1. From the Alpha Five Control Panel’s Table/Sets tab, double-click the Product table to open the default browse. Navigate to the last record and note that last product that was entered. You can leave this window open.

2. From the Alpha Five Control Panel’s toolbar, open the Interactive Window.

Type each line into the Interactive Window and press ENTER.
3. Type each of the following lines exactly as shown into the Interactive Window, pressing ENTER after each line.

```xbasic
dim tblProd as p
tblProd = table.open("product")
tblProd.enter_begin()
tblProd.Description = "My product"
tblProd.Cost = 1.99
tblProd.Retail = 2.99
tblProd.Vendor = "V007"
tblProd.enter_end()
tblProd.close()
```

4. Now, go back to the Product table default browse. Press F5 Refresh to show the new record.

![Product table default browse](image)

After pressing F5 to refresh the Product table default browse, you will see the new record that you just entered.

Note that you were able to add the record with the browse open, but that you had to refresh the browse to show the new record.

In this exercise, you:
- Used the Interactive Window.
- Used Xbasic to open a table, enter a record and then close the table.

**Variables**
EXERCISE 2

In this exercise, you will see how to set variables and use them in Xbasic.

1. Return to the Interactive Window. Clear it by choosing Interactive > Clear Window from the menubar.

5. Create a variable named vDescription. It is good practice to dimension the variable first with a DIM statement.

Type each of the following lines exactly as shown into the Interactive Window, pressing ENTER after each line.

dim vDescription as c
vDescription = "My variable product"
?v Description

The first line dimensioned the variable, giving it a name and type “character”. The second line assigned a value to the variable. The third line returned the value of the variable to the screen. The “?” is valuable in the Interactive Window for testing purposes. It is also used in Alpha Five web pages to display the contents of a variable.

6. Now that you have created the variable vDescription and assigned it a value, you can use that variable in your Xbasic code.

Type each of the following lines exactly as shown into the Interactive Window, pressing ENTER after each line. Alternatively, save time by copying and pasting this code into the Interactive Window. Then select the lines with your mouse and choose Interactive > Run the Selected Text from the menubar.

    dim tblProd as p
    tblProd = table.open("product")
    tblProd.enter_begin()
    tblProd.Description = vDescription
    tblProd.Cost = 1.99
tblProd.Retail = 2.99
tblProd.Vendor = "V007"
tblProd.enter_end()
 tblProd.close()

7. Browse the Product table to see the new record that you entered.

In this exercise, you:
- Created a variable and assigned it a value.
- Used the variable in Xbasic code.

Using Action Scripting to Generate Xbasic Code

EXERCISE 3

This exercise shows how Alpha Five’s Action Scripting genies can be used to generate Xbasic code. You can use Action Scripting as an aid to learning and writing Xbasic.

The example uses a button on the Products by Vendor form. In this exercise you will not create the button from scratch; rather, you will use a pre-written Action Script and learn how to view the Xbasic behind Actions and convert Actions to Xbasic.

1. Open the form named frmVendorProd in Form View. Try the Add a Product button to see what it does. Recommended: Use Atlanta Sporting Goods for testing.
2. Open the form named frmVendorProd in Design View. Right click the “Add a Product (Action Scripting)” button and choose Events > OnPush.

3. Look at what each Action is doing. Select an action with your cursor and choose Edit Action. Then step through the Script Genie by clicking Next. Click Finish or Cancel to close the Script Genie when done.

   The first action, Display an Xdialog Box prompts the user for the Description, Cost and Retail and stores their responses in variables.

   The second action, Set variable(s) to field values, gets the Vendor ID from the form and stores it in a variable.

   The third action, Xbasic Enter Record, does the same thing we did in the previous exercises. In this case it uses the variables that were set by the first two actions.
The fourth action, Refresh Object, does something similar to F5 Refresh but only on the specified object – the embedded browse on the form.

4. To view the Xbasic behind an Action, right-click the Action and choose View Xbasic.

5. To convert an Action to Xbasic, right-click the Action and choose Convert to Inline-Xbasic.

Convert the Xbasic Enter Record action to Xbasic by right-clicking the action and choosing Convert to Inline-Xbasic. A message will warn you that you cannot undo the conversion. Answer “Yes” to proceed.

6. To view the converted Action, right-click on it and choose Edit Action. Note that the code uses variables such as vDescription that were created in the Actions preceding the Xbasic Enter Record action.
7. Back out of the script editor by clicking Cancel. Close the form without saving it.

In this exercise, you:
- Saw how variables can be set and how they are used in Xbasic.
- Converted an Action Scripting action to Xbasic.

**Bonus Example:** You can convert an entire Action Script to Xbasic. For an example, check out the three “Print Report” buttons on the Product by Vendor form.

### Conditional Statements

**EXERCISE 4**

In this exercise, you will modify a pre-written Xbasic script to add a Conditional Statement. The example uses the “Add a Product (Script to add record)” button on the Products by Vendor form. The button runs the script named scp

1. Open the form named frmVendorProd and click the Add a Product (Script to add record) button. Try it to see what it does. Recommended: Use Atlanta Sporting Goods for testing.
2. Open the form frmVendorProd in Design View. Right-click the Add a Product (Script to add record) button and choose Events > OnPush. Note that the third action runs a script called scpProductAdd.


4. From the Alpha Five Control Panel, Code tab, open the scpProductAdd in Design View.

Note: To turn on line numbers, right-click in the Code Editor and choose Line Numbers.
5. Modify the scpProductAdd script to add a conditional statement. This IF statement will calculate the retail price if the retail price was left blank.

At line 25, modify the code so that it appears as shown below.

![Code Editor]

IF...ELSE...END IF is the most frequently used conditional statement. For multiple conditions, you can use IF...ELSE...ELSEIF...END IF. Or you can use the SELECT...CASE...END SELECT statement.

Tip: Indenting is optional but makes IF statements easier to read, especially when nested. Alpha Five will indent your code for you. In the Code Editor, choose Code > Format from the menubar.

6. Choose File > Save from the menubar to save the scpProductAdd script.

7. Open the form frmVendorProd in Form View and click the Add a Product (Script to add record) button. This time, leave the retail price blank. When you click OK, you will see that the retail price was calculated by the script.
In this exercise, you:
- Used the Code Editor.
- Added a conditional IF statement to a script.

EXERCISE 5

This exercise demonstrates how Xbasic is used in a dialog component in Web Projects. The dialog component will enter a record into the Product table.

In this exercise you will not create the dialog component from scratch; rather, you will open and review a pre-written component.

1. From the Alpha Five Control Panel, choose Web Projects.

2. In the Web Projects Control Panel, choose Web Components. Edit the component named dlg_ProductAdd.
3. First, preview the dialog component in the browser so that you can see what it does.

In the Dialog Builder, choose Browser from the toolbar. If the Application Server is not started, you will be prompted to start it.

4. Your browser will open a Live Preview of the dialog component. Fill out the form and click Submit. Recommended: Use Atlanta Sporting Goods for testing.

In Live Preview, choose the Vendor and fill in the Description and Cost. Click Submit.
5. Return to the dlg_ProductAdd component in the Dialog Builder. Choose Controls.

The controls on the dialog box will be used as variables in the script that adds a new product when the user clicks the Submit button.

6. In the Dialog Builder, choose Properties. The script that adds the new record to the Product table is on the dialog component’s After Validate event.

Edit the After Validate event to view the code. The Xbasic can be written (or
generated by Action Scripting) and tested on the desktop, then pasted into the Form Events editor.

Some modification is required for the code to run successfully.

7. Back out of the Form Events editor and the Dialog Builder by clicking Cancel. Close the dialog box without saving it.

In this exercise, you:
- Saw how Xbasic can be used in web components.

**Text String Manipulation**

**EXERCISE 6**
Alpha Five comes with many built-in functions for manipulating text. Some of the most useful are: ALLTRIM(), WORD(), LEFT(), RIGHT(), SUBSTR(), STRITRAN() and AT(). See also the Alpha Five Help topics: Character Functions and List Processing.

In this exercise you will create your own user-defined function to convert a name “firstname lastname” to “lastname, firstname”.


2. Create a new function named fncLastFirst with the optional argument parmName.
3. Click OK to go to the Code Editor. In the code editor, type the following lines between the FUNCTION and END FUNCTION lines:

```xbasic
dim vFirst as c
dim vLast as c
vFirst = word(parmName, 1)
vLast = word(parmName, 2)
fncLastFirst = vLast + "," + vFirst
```

4. Save the function by choosing File > Save.

5. Go to the Interactive Window to test the function that you just created. Type:

```xbasic
?fncLastFirst("any name")
```

In this exercise, you:
- Created a user-defined function.
- Used the built-in Alpha Five WORD() function.
- Wrote an expression that manipulated text.

**Looping**

**EXERCISE 7**

This exercise shows how to loop through records in a table with the WHILE...END WHILE statement. The example is a pre-written function that builds a text string from values in those records.

1. First, open the form named frmInvoice and look at Invoice Number 00002 to see which items were purchased.
2. Go to the Interactive Window. Type `fncInvoiceItemString("000002")` and press ENTER.

3. The function builds the string based on how many items are in the invoice. Try passing different invoice numbers to `fncInvoiceItemString()`.

4. Open the `fncInvoiceItemString()` in the Code Editor. This user-defined function is heavily commented to explain exactly what it is doing at each step and provide tips for the novice Xbasic programmer.
The essence of the function’s WHILE…END WHILE loop through the items on the invoice is:

```vbnet
tbl = table.open("invoice_items")
query.filter = "Invoice_Number = " + parmInvoice_Number + ""
query.order = ""
query.options = "T"
ndx = tbl.query_create()
tbl.fetch_first()
WHILE .not. tbl.fetch_eof()
    ' Code to be performed on each record goes here.
    tbl.fetch_next()
END WHILE

tbl.close()
```

In this exercise, you:
- Saw the WHILE…END WHILE looping statement and SELECT…END SELECT conditional statement in action.
- Learned how to read through records in a table.

**Bonus Example:** The fnclInvoiceItemString() function can be incorporated into a customer-friendly letter, e-mail or on-screen message. For an example, check out the “Follow-up E-mail” button on the Invoice form.
**Converting Saved Operations to Xbasic**

Alpha Five provides many types of Operations for manipulating and updating data. After creating an Operation, you can save it and easily re-run or modify the saved operation later.

There are two reasons to convert a Saved Operation to Xbasic:
- To add variables and conditional statements.
- Web applications cannot run Saved Operations.

Fortunately, Alpha Five makes it easy to convert saved operations to Xbasic.

**EXERCISE 8**

In this exercise you will convert a Saved Export Operation to Xbasic and modify it to prompt the user for the export file name at run time.

1. From the Control Panel’s Operations tab, open the operation oprProductExport in Design View.

2. Click the XB “Show Xbasic Code” icon on the toolbar.

3. You will have several options. The Processed options contain additional code that is worthy of study, but for this exercise, choose “Raw”. Click the “Save As Script” button. Save the script with the name “scpProductExport”.

![Click the XB “Show Xbasic Code” icon on the toolbar.]

![Alpha Five - [Product.db : Expert [oprProductExport]]](image)
4. Close the Xbasic window and close the Export operation without saving.

5. In the Code Editor, open the pre-written script “scpProductExport1” in Design View. The script scpProductExport1 is a slightly modified version of the script that you just saved. The table.open("product") and table.close() have been added for you.

6. Place your cursor on Line 8 of the scpProductExport1 script. Click the Xbasic Script Genie icon on the toolbar. Choose the Prompt for a Filename action.

7. Click OK to proceed to the Script Genie. Fill in the values as shown below:

   File Filter: CSV Files (*.csv)
   Variable Name: vExportFile
   Dialog Title: Product List Export File
   Default Filename: product_list.csv
   File Selected: Must Not Exist
8. Click Finish. The next window will show you the Xbasic code for the Prompt for a Filename action. Click the “Insert Code into Script Editor” button.

9. In the Code Editor, change the line that specifies the name of the export file for the operation, at around Line 34 of the script.

Change:

```plaintext
export.file = "product_list.csv"
```

To:

```plaintext
export.file = vExportFile
```

The vExportFile variable will contain the file location and name that the user selected from the filename prompt.

10. Save the scpProductExport1 script and close the Code Editor. Test the script by running it from the Code tab on the Alpha Five Control Panel.

    Note: The pre-written script scpProductExport2 is an enhanced version of scpProductExport2. It notifies the user when the export is finished and gives them the option to view the export file.

In this exercise, you:

- Converted a Saved Operation to Xbasic.
- Used the Xbasic Script Genie to generate code in the Code Editor.
- Prompted the user for a filename.
Web Applications Genies

The code editors in the Web Projects Control Panel contain a limited number of genies to help you write Xbasic. In the Component Builders’ Events editors you can find Genies by clicking the Insert button. When inserting Xbasic into A5W pages right-click in the editor window and choose Genies.

EXERCISE 9

In this exercise you will use the Report Print Genie to generate Xbasic code for use in a dialog component.

1. First, take a look at the Products By Vendor report. From the Alpha Five Control Panel’s Reports tab, select the report rptProdByVend and choose Preview.

2. Close the report preview. From the Alpha Five Control Panel’s toolbar, click the Web Projects icon to open the Web Projects Control Panel.

4. Go to the dialog component’s Form Properties, scroll down to the After Validate event and click the button to edit.
5. The Form Events editor will be empty. Click the Insert button and choose Genies > Print Report. The Print Report Genie will open. Choose the report rptProdByVend.

6. Add a filter variable named vVendor_ID of type “character”. The variable name matches the name of the dialog component’s DropDownBox control.
7. In the Print Report Genie, select the report rptProdByVend.

Add a variable vVendor_ID, to be used in the Filter Expression. The Filter Expression will filter the report by Vendor, based on the user’s selection from the drop down box on the dialog component.

Use the Filter Builder to select the following. See the screenshots below for the steps.

Field: Vendor_ID
Operator: Is Equal To
Variable: vVendor_ID
8. Click OK to insert the variable in the Filter Builder, then click OK again to insert the filter expression into the Print Report Genie.

"Vendor_id = \""+vVendor_ID+\"\"

Writing a filter expression can be very challenging and troubleshooting errors can be time-consuming in a web application. The Filter Builder helps you get the syntax right.

9. For this exercise, the Order Expression can be left blank and the Report Type should be “PDF”. Click OK to complete the Print Report Genie. The code it generated will appear in the Form Events code editor.

Note that the genie substituted the path alias [PathAlias.ADB_Path] for the location where the report is stored on the local computer. This will allow the web application to find the report on the server. Also note that the PDF file is being output to a temporary file in the session folder so that multiple users/sessions can print at the same time.
10. Click OK to return to the Dialog Builder. Save the dialog component.

11. Preview the dialog component in the Browser.

12. The dialog component will appear in your browser. Choose a vendor and click Submit to print the report.
In this exercise, you:

- Used the Print Report Genie to generate Xbasic in a dialog component.

**Bonus Example:** The web component `dlg_ProductOutput.a5wcmp` and page `ProductReport.a5w` combine the export from EXERCISE 8 and the report from EXERCISE 10. The dialog gives the user the option to output the product list to a report in PDF or to a file in CSV. For details and special features refer to the comments in the component’s After Validate event and in the page.

**Xbasic Tips**

- Name objects in your database (tables, layouts, operations, scripts, web components and pages) using only letters, numbers and underscore. Spaces and symbols will make it harder to refer to these objects in your Xbasic code.

- Name variables with a prefix such as “v” so you won’t confuse them with fields and other objects in your code.

- By default, variables in Alpha Five have a local scope, meaning that they only exist within the current script or web page. Action Scripting scopes variables as “shared”, meaning that they persist while the form is open. Sometimes you will need to scope a variable as “global” so that the variable and its value will be
available everywhere in Alpha Five. Global variables persist until the database is closed. To help you keep track of them, give global variables a special prefix, such as “vg”, and use descriptive names. Use global variables sparingly.

- Build and test your code in pieces. When you get stuck, try a simplified version of what you are attempting to do.

- Use comments liberally. In the Alpha Five code editors, anything that comes after an apostrophe on a line will be treated as a comment.

- Use the Xbasic Script Genie to generate and insert code into the Code Editor.

- Use Format Code to organize and indent your code in the Code Editor.

- Some Xbasic can be used only in desktop applications or only in web applications. Check the command’s Alpha Five Help topic for “Limitations”.

More to learn on your own

Here are some topics that were not covered but are worth looking up in Alpha Five Help or on the Alpha Message Board.

- Changing (updating) records with Xbasic
- Fetching through records
- List Processing
- FOR…NEXT looping statement
- Error handling for Xbasic on web and desktop

The Action Script actions contain a wealth of techniques. Study the Xbasic behind them.

Questions? Comments?

Contact Frances Peake at Proctor & Peake, Inc.
(305) 751-1181 frances@proctorandpeake.com