

## Chapter 6.

# *Take it to the world!*



### *There's no limit ...*

Do you fancy an application suitable for the web or mobile – or maybe both? Whatever your aim, Alpha Five will get you there. Applying scripts is a bit different than for the desktop, but the code writing principles are the same.

*What's in this chapter*

Topic	Desktop	Web/Mobile	Page
<b>Background</b>			
“Begin at the beginning”			145
“Hand coding in web applications”		✓	145
<b>Writing and placing the code</b>			
“Understanding the Code section”		✓	153
“Writing JavaScript for button and hyper-link events”		✓	158
“Understanding Xbasic property settings”		✓	160
“Understanding JavaScript property settings”		✓	163
“Using grids on the desktop”	✓		160
“Creating Server-side and Client-side Events”		✓	170
<b>Saving the code for future use</b>			
“Creating an AEX Library”	✓	✓	172
“Creating a function and saving it to the AEX Library”		✓	176
“Stored Procedures”		✓	152

*Preparation for the exercises*

This chapter continues with the *Phone Messages* workspace that ships with Alpha Five. If you have not already done so, please follow the instructions on page 8 to place a copy on your computer.

The initial instructions reference the “HelloWorld” grid component that was developed at “Creating a message box with Action JavaScript” on page 138. Please complete that exercise before beginning this chapter.

**DESKTOP USERS TAKE NOTE**

This chapter and the next will focus on web applications, but remember that most web components can be used on the desktop, too.

## *Begin at the beginning*

---

If you are new to Alpha Five programming and have jumped here without reading the rest of this book, we really recommend that you head back to Chapter 1.

**Here's why.** Even though the early chapters give examples of programming in desktop applications, the same principles apply to programming on the web. It is important that you understand objects, variables and other design principles before beginning this chapter.

**And.** Even if you understand those concepts, be sure you understand the functionality of the Alpha Five Code Editor and the Interactive Window. You also need to know how to create Functions in Alpha Five.

- “Understanding the Code Editor” on page 10.
- “Understanding the Interactive Window” on page 17.
- “Creating Global (User Defined) Functions” on page 99.

## *Hand coding in web applications*

---

The most important element to understand when coding in web applications is which type of code – SQL, Xbasic or JavaScript – is appropriate and where it is placed. To a large degree, that depends on whether you are working on the “client-side” or the “server-side.” We’ll explain those terms in a moment, but for now, be sure to remember this when you are using Alpha Five for web applications:

- The client-side uses JavaScript.
- The server-side uses Xbasic and/or SQL.

JavaScript is the language that the browser understands. Alpha Five uses Xbasic on the server-side because it is extremely flexible and has been designed to meet the needs of the program and make coding faster and easier. Alpha Five also has portable SQL and supports native SQL for most programs.\*

- Developing custom code for the web is a constant back and forth between JavaScript and the server-side code.

### TERMINOLOGY

There are some terms that you will hear often in discussions about Alpha Five web application programming – JavaScript, AJAX and Event Handlers. At the top of the list, however, is being very clear on the terms, “client-side” and “server-side.”

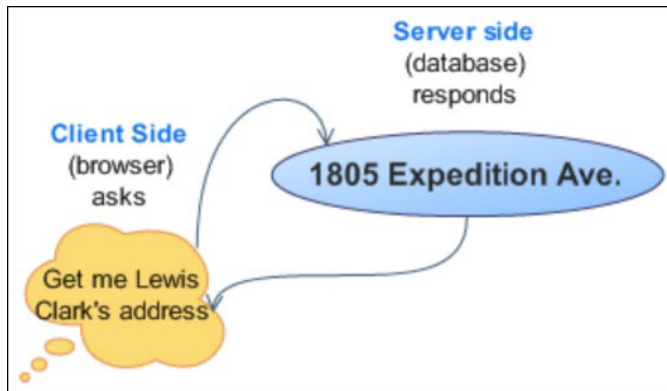
---

\*. See “Portable SQL” on page 152.

## Understanding the Client / Server model

The biggest difference between developing for web applications as opposed to those for the desktop is the concept of Client and Server.

“Client/server describes the relationship between two computer *programs* in which one program, the client [your browser], makes a service request from another program, the server [your database files], which fulfills the request.



Understanding this concept is important. As in any field, the terminology is key. Don't confuse this relationship with “server” hardware. In this case, we are talking about software programs. The client is your browser and the server is the database. Both are software products. When you think of it that way, you can begin to see why HTML and JavaScript are required for the translation.

“Although the client/server idea can be used by programs within a single computer, it is a more important idea in a network. In a network, the client/server model provides a convenient way to interconnect programs that are distributed efficiently across different locations.\*

CHECK YOUR BANK BALANCE -  
2 ROUND TRIPS

“Computer transactions using the client/server model are very common. For example, to check your bank account from your computer, a *client program* in your computer forwards your request to a *server program* at the bank. That program may in turn forward the request to its own client program that sends a request to a database server at another bank computer to retrieve your account balance. The balance is returned back to the bank data client, which in turn serves it back to the client in your personal computer, which displays the information for you.

CENTRAL IDEA IN NETWORK  
COMPUTING

“The client/server model has become one of the central ideas of network computing. Most business applications being written today use [this] model....

MANY CLIENTS - ONE SERVER

“In the usual client/server model, one server, sometimes called a daemon, is activated and awaits client requests. Typically, multiple client programs share the services of a common server program. Both client programs and server programs are often part of a larger program or application.

WEB BROWSER IS CLIENT  
PROGRAM

“Relative to the Internet, your Web browser is a client program that requests services (the sending of Web pages or files) from a Web server (which technically is called a Hypertext Transport Protocol or HTTP server) in another computer somewhere on the Internet.”†

\*. In this case, network refers to Internet or Intra-net connections, not to local computer connections that use an Ethernet cable or internal wireless connections.

†. **Client/Server definition:** SearchNetworking.com; [http://searchnetworking.techtarget.com/sDefinition/0,,sid7\\_gci211796,00.html](http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci211796,00.html), Jan. 23, 2008.

This Wikipedia\* chart shows an analysis of clients and servers:

Characteristics of a client	Request sender is known as client
	Initiates requests
	Waits for and receives replies.
	Usually connects to a small number of servers at one time
	Typically interacts directly with end-users using a graphical user interface
Characteristics of a server	Receiver of request which is sent by client is known as server
	Passive (slave)
	Waits for requests from clients
	Upon receipt of requests, processes them and then <b>serves</b> replies
	Usually accepts connections from a large number of clients
	Typically does not interact directly with end-users

## XBASIC AND THE WEB

Now that you understand how applications designed for use on the web differ from those for the desktop, let's translate that into what you can and cannot do without using JavaScript.

- An Alpha Five Xbasic<sup>†</sup> user asks, “How can I use Xbasic on the Alpha Five page to capture a value changed on the client browser by JavaScript without making a request to server.”
- The answer is he can't, because **Alpha Five code only exists and works on the server**. Client side actions must be done in JavaScript.

Other examples needing JavaScript are dynamic lookups and capturing actions and events.

You can, however, use Xbasic to enhance many areas of your web application. You can use it to add features to a web page. You can write Dialog Event Scripts. You can write Server Event Scripts.

## A5 OPERATION

You cannot run a saved operation directly from an A5W<sup>‡</sup> page or web component, however, you can copy its Xbasic code and place it into a script in an A5W page.

## THE GOOD NEWS

The folks at Alpha Five keep making the web easier and easier to use. There are builders for web components — you just pick from the options. There are

\*. **Client / Server:** <http://en.wikipedia.org/wiki/Client-server>, Jan. 23, 2008

†. Xbasic is Alpha Five's programming language.

‡. A5W pages are created with Alpha Five's built-in HTML editor.

features, such as Action JavaScript and JavaScript based date pickers and look-ups. There is brilliant security. And lots more.

**But.** They can't provide everything because they can't change the HTML standards or the limitations of browser technology. Regardless of the range of devices they have provided — and you'll find they are truly amazing — those wanting to build to a high level of sophistication, today, will need to learn JavaScript.



ANOTHER IMPORTANT NOTE!

In Alpha Five, the separation between client-side (done in Javascript) and server-side coding (done in Xbasic/SQL) is not always as clear as one might think because there are certain client-side events that use Xbasic.

**But, you ask, “How can that be?”** Action JavaScript is one answer. In certain situations, such as the *onClick* event for a button – a client-side event – you can write an Xbasic function that talks back and forth between the browser and the server – all done by Alpha Five behind the scenes.

**And.** There are Xbasic functions that convert data to browser standards. Kudos to its flexibility!

**Be patient.** All will be revealed in the upcoming exercises. But first, there are some more important terms to learn.

## What is JavaScript

We just mentioned you might want to learn JavaScript. What's that?

“JavaScript (sometimes abbreviated JS) is a scripting language commonly implemented as part of a web browser in order to create enhanced user interfaces and dynamic websites.”\*

The important thing to know here is that, when programming for the client-side, JavaScript is required.

**And.** That Alpha Five makes it all very easy by supplying *Action JavaScript* that will handle many of your needs.

**And.** That you can also write your own JavaScript when you want to.

## JSON

Javascript has a sub-language called JSON (JavaScript Object Notation). It is “a lightweight data-interchange format that is easy for humans to read and write and easy for machines to parse and generate.”†

Alpha Five has functions that allow you to take advantage of this language, too. For example, the *vartoJSON()* function converts an Xbasic variable to a JSON representation.

---

\*. <http://en.wikipedia.org/wiki/JavaScript>. 11.28.12.

†. <http://www.json.org/>

---

DON'T PANIC!

You have to crawl before you can walk, so tuck JavaScript into the back of your mind for now. You can pick it up later – after you see how it fits into Alpha Five.

## What is AJAX?

Ajax is basically a lightweight way for a browser to exchange a little data with the server without having to refresh the whole page. It smooths and speeds access to your application data.

Regardless of file format – SQL or DBF – data in web applications must travel from the end user's computer to the server where the data resides. Each information request and data change requires trips back and forth between the two. This takes time and can produce a jerky appearance as the screen refreshes.

- **AJAX** (Asynchronous JavaScript and XML) smooths and speeds the transition of data because it changes the way data is accessed from the server so web databases behave almost like those for the desktop. It's been around a while, but has always required advanced programming knowledge to add it to an application.
- The terms are defined as follows:
  - **Asynchronous:** “In programming, asynchronous events are those occurring independently of the main program flow.”\*
  - **Javascript:** See above.
  - **XML:** “Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.”†

**Alpha Five gives you AJAX without coding!** The folks at Alpha Software have done all the work for you

**So.** You can just go about the business of creating your web application. You don't have to know JavaScript or even how the motor runs. It's just in there.

**But.** There are times when you may want to define your own AJAX call backs. We'll explain that in a moment.

## What is best? SQL or DBF???

Alpha Five V11 has a free “Feature Pack” that converts Alpha Five files from DBF to SQL. That brings up the question of who should make that switch and why.

I have always been a great fan of the desktop workspace side of Alpha Five for its ease of use for the non-programmer and readily admit to my lack of knowledge about SQL. Therefore, I turned to Jerry Brightbill of Alpha Software, an expert programmer in both file types. Here is a synopsis of our e-mail exchanges:

---

\*. <http://en.wikipedia.org/wiki/Asynchrony>. 12.1.2012

†. <http://en.wikipedia.org/wiki/XML>. 12.1.2012

---



MY QUESTION

“I’m adding notes on the new *Upsize Database Genie Feature Pack*\* to the book. I understand that SQL is preferable to DBF for large databases.

- Would you give me a brief explanation of why someone might want to make the change, keeping in mind that many of my readers are ‘small guys.’
- How large (number of records) does your database have to be in order to see a significant change in performance between SQL & DBF.
- What other factors enter into the decision?
- Once the conversion is made, will the developer need to know how to create SQL Select Statements, etc.?”

JERRY’S RESPONSE

“I have been using SQL (SQL Server) exclusively for a couple years, so I may be a bit biased. Frankly I can’t see why anyone would use DBF in most situations except that it comes with Alpha Five. But, to be fair, there are pros and cons for each. This article gives a somewhat accurate evaluation of both:

<http://www.foxdev.com/VFPonSQLSGeneric.pdf>

“However, it does have a few contradictions.

LOCAL VS REMOTE AND SLOW ACCESS

“DBF can be faster when the data is local, but the speed difference begins to fade when the data is remote and file size and the number of users increase. The reasons are simple.

- “The DBF opens the entire table to get a couple records. Depending on the file locking algorithm, some or all records could be locked to other users.
- “All flavors of SQL just query and pull a subset of data based on the query. Once that result set is obtained, the table is closed and available to another user. Even a query on a very large table can be very fast if the number of columns and rows is limited.

“It is very interesting that the article touts DBF speed as a pro, and yet slow access for DBF is a con. Aren’t they the same thing?”

PROS AND CONS

“What I find revealing is that DBF has only three benefits listed, but five limitations. SQL has five benefits and only two limitations.

COST

- “The cost issue for SQL is interesting as *Microsoft SQL Server* and *MySQL* have free versions that may suit the needs of many users. Yet, if there is a need to expand and add features, both of them and others have very robust (and expensive) expanded versions.
- “For example, I am working with a client on a system using SQL Server. They are using *SQL Server Enterprise* 64 bit because of built in data encryption, multi-processor and drive support, and other features for scalability. It has very powerful import and export capabilities and a report writer. Because the server can use all available resources in very efficient ways, its speed is dramatic. So is its price. A single server license is well over \$10,000. But for their purposes, it is exactly what they need.

COMPLEXITY

“The complexity argument doesn’t carry much weight with me either. The programming language needed to interface with SQL is different, so there is a learning curve.

- But it is so much more robust and capable that the perceived complexity isn’t a major issue.

---

\*. An optional V10 Feature Pack that has been included with V11 at no extra charge.

---



- Much program logic can be moved from the application layer to the database layer for increased performance.
- DBF basically has little or no capability for data manipulation at the database level. Almost everything has to be done in the application layer. The exception in Alpha Five is the application of field rules. That is done at the table level.

“This presents another advantage of most SQL solutions not mentioned. When much of the data logic is placed at the database level, many different applications can interface with that data without having to duplicate process logic. Applications become ‘front-ends’ for data and a lot of data processing is done on the server.

- “SQL can be much more complex, but it can be relatively simple as well.

## CONCLUSIONS

“In my mind, if you are building a system that requires data integrity, security, and scalability, there is no contest. DBF just doesn't stack up well against any SQL database in any of those areas.

“If simplicity and cost are the only criteria, then DBF is still a solid product and it has its uses.

- “Support for SQL in Alpha Five is getting very good, particularly on the web and in Xbasic. The Active Link tables on the desktop aren't quite as efficient, but they are getting much better.”
- If you think the time has come to consider converting your Alpha Five database to SQL, check out the “Database Upsize Genie” on page 250.

**But.** Keep in mind that we use DBF in many of our exercises because it comes with the package and does not require additional software. Both formats behave the same for our exercises.

**So.** You can hold that decision for now, if you like.\*

## What is an event handler?

“In computer programming, an event handler is an asynchronous callback subroutine that handles inputs received in a program (called a listener in Java). Each event is a piece of application-level information from the underlying framework, typically the GUI toolkit. Graphical User Interface (GUI) events include **key presses, mouse movement, action selections, and timers expiring**. On a lower level, events can represent availability of new data for reading a file or network stream. Event handlers are a central concept in event-driven programming.” †

Alpha Five uses event-driven programming, therefore you will hear the term, event handler, used frequently – especially in the videos. We have already explored Alpha Five desktop events, such as the *OnPush* event for a button and we took a quick peek at the *onClick* event for a grid button on page 138. Upcoming exercises will expand on events.

---

\*. Reprinted from Alpha Five Web Applications Made Easy, The Basics and More for V11, Susan Hussey Bush; pp. 14-6.

†. [http://en.wikipedia.org/wiki/Event\\_\(computing\)#Event\\_handler](http://en.wikipedia.org/wiki/Event_(computing)#Event_handler) 11.28.2012.

---

## Understanding how to write the code

### XBASIC

Unlike the desktop where one creates a global function only when it will be used repeatedly, in web applications, **all** scripts are created as Functions.

- See “Creating Global (User Defined) Functions” on page 99.

To use one repeatedly, it is placed in a Aex Library.

- See “Creating an AEX Library” on page 172.

### PORTABLE SQL

The folks at Alpha Five have developed a “portable SQL” syntax to get past the problem that not all SQL is the same. Alpha Five invisibly translates portable SQL to native SQL for the particular program, so it works across all database platforms.

With V11, native SQL – the specific syntax used by the particular SQL database (MySQL, Oracle, SQL Server, etc) – is also supported. So in almost all cases, Alpha Five can support any behavior that is produced by the back end database.



HINT

If you are using a SQL database, the more experience you have working with it directly, the better. Even if you are an expert, however, be sure to investigate Alpha Five’s portable SQL and other SQL functions.

- Documentation > Filter: Enter **Portable SQL**. Choose:
  - Portable SQL and Supported Portable SQL Syntax.
- Documentation > Filter: Enter **SQL**. Scroll down for list of functions.\*

## Understanding where to write the code

Web applications have several code builders. In many cases, you will find them right where you need them. As a guide, when you see an Interactive window tab, you can write your code on the spot. You can also write code on .A5W pages.

### WEB COMPONENTS

Web components also have designated areas for quickly writing/reviewing Xbasic/SQL and Javascript.

### AEX LIBRARY

If you plan to save an Xbasic Function to an Aex Library, write it in the Code Editor.

### STORED PROCEDURES

Most SQL databases support Stored Procedures. In effect, these are somewhat equivalent to Xbasic functions. They accept input parameters and return some output, giving the developer the opportunity to write some process and save it.

Stored Procedures are created in the SQL database and referenced in Alpha Five. Because they run natively in the actual database, they can be orders of magnitude faster than Xbasic. We know of one instance where the major tables contain well over 1 million records each. It has a view consisting of 22 tables

---

\*. All Portable SQL and other SQL functions are listed in the Functions Directory, available at [www.libertymanuals.com](http://www.libertymanuals.com).

linked together in various ways. It's developer reports that, "While it is slower than a single table, it isn't very much slower."

Previously, you had to create a Passive-Link Table in order to use a Stored Procedure. That changed in V11 and Stored Procedures in updateable grids are now permitted.



GO TO A5 HELP (AND VIDEOS)

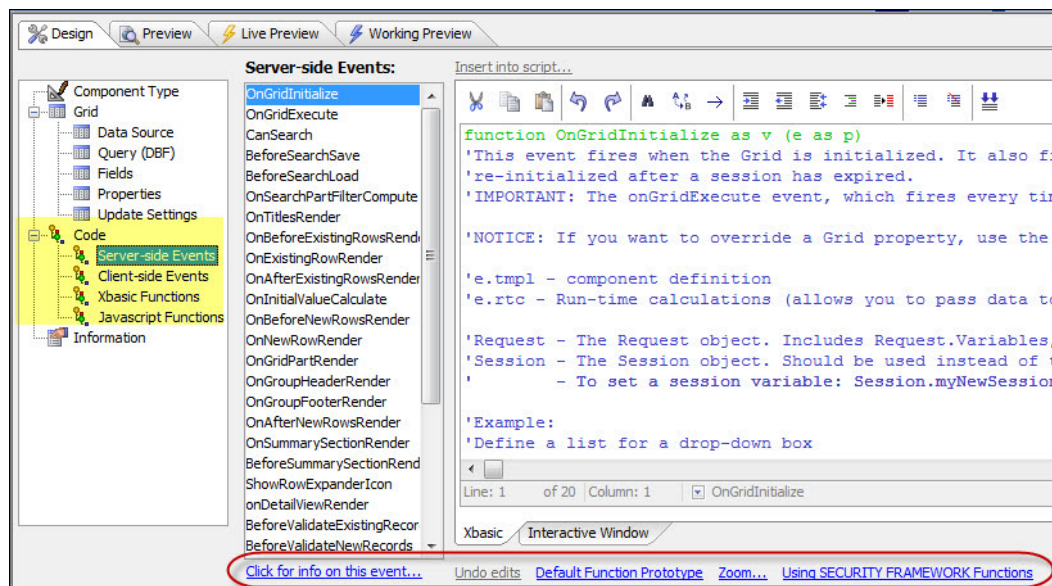
The first page below has links to two videos with clear explanations on using Stored Procedures in Grids.



- Documentation > Filter: Enter **Stored Procedure**. Choose:
  - How to call a SQL Server stored procedure from a Grid 10.5. **2 videos**.
  - Also shows using the Debugger in a manner that we will detail in the next chapter.
  - Web Grids based on a Stored Procedure.

## Understanding the Code section

The Code section of the Grid Component has four places where Xbasic and/or Javascript may be entered.



1. Open the *HelloWorld* grid component that we created in "Creating a message box with Action JavaScript" on page 138.\*
  - If you have not done that exercise, we suggest you do so now. It will help in understanding this chapter.
2. *Navigator*: Refer to the **Code** section (yellow above).

\*. See "Preparation for the exercises" on page 144.