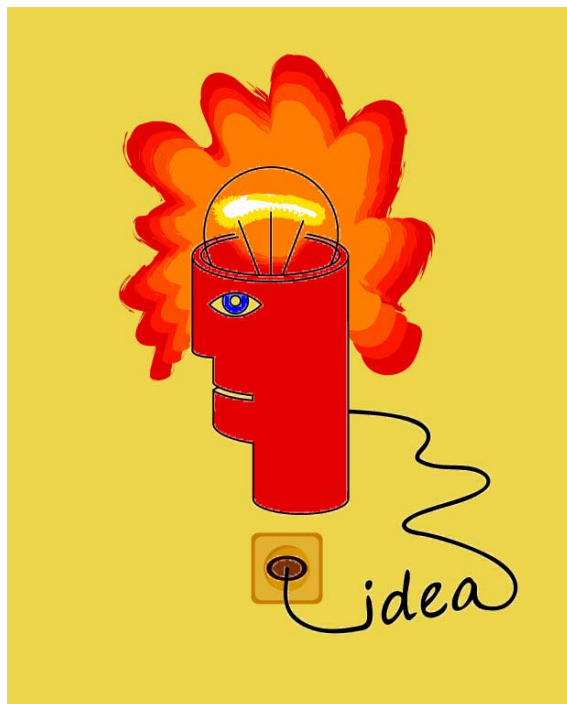


Chapter 3.

Powerful Prodigy



The idea that makes mobile development explode...

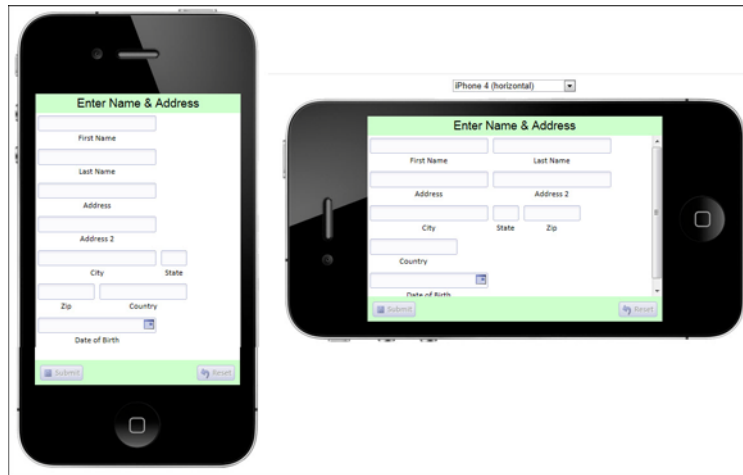
Discover how the UX component makes mobile application development possible in an entirely new way. And - you can develop once for all types of devices. And - it also serves as a dialog for web apps.

How brilliant is that!

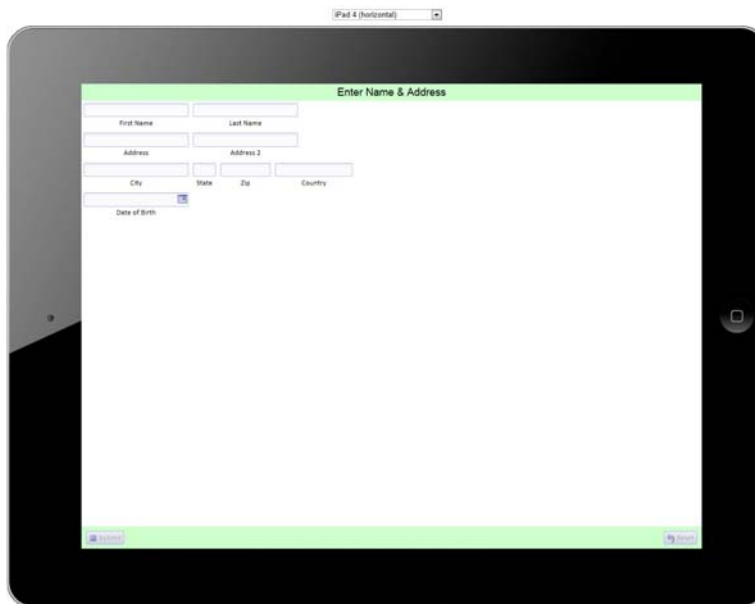
Mobile Device Simulator

The UX Component has an exciting feature that will have you longing to create Mobile applications. This chapter will show how to create these sweet babies!*

Phone



iPad



*. iPhone and iPad are trademarks of Apple, Inc. This chapter is reprinted in part from *Alpha Anywhere Made Easy, The Basics and More for the Desktop* and *Alpha Anywhere Report Writer, Reporting and Charting Made Easy* by Susan Hussey Bush, Liberty Manuals, 2013.

How the material is organized

The previous chapter gave an overview of how the Alpha Anywhere system works. Now it is time to look at a very important component that serves two purposes. It acts as a dialog and as the foundation for mobile design.

- Dialogs ask the end user for information – like the name and address info you fill out when you buy something on the web.
- Mobile requires small data packets because transmission and device speed are often slower than in web applications. The UX is very efficient and, in most cases, replaces the Grid in mobile apps.

Those of us who have been around a while, remember when the Grid component was *the* revolutionary concept in web application design. Well, now it's time to cheer for the UX, *today's* revolutionary concept in mobile application design!

The UX Component sets mobile application design apart from the competition. Nobody has anything like it. Spread the word!

This chapter will explain how to use the UX for data entry and submission to a table. Then we will display a list of records in a table and use it to filter a report. We will also create a menu, show how to use the mobile device simulator and how to test the component in your smart phone or tablet.

ACTIONS SCRIPTING & ACTION
JAVASCRIPT

Alpha Anywhere provides two tools that write code for you. *Action Scripting* writes Server-side Xbasic, while *Action Javascript* writes Client-side Javascript. We will use both in this chapter.*

PANELS

Of all the topics discussed in this chapter, Panels are the most important for **anyone even considering mobile application development**. So important that we recommend you stop right now and watch these videos.



GO TO ALPHA VIDEO

- Videos > Filter: Enter **Panel**. Choose:
 - #3. Tutorial Explaining panel Layouts, panel Navigators and panel Cards. 5 videos.

FILE TYPE CHANGE

Certain exercises require data be entered in a table. In those situations, we will switch from the Alphasports SQL table that we used in the previous chapter to an Alpha Anywhere DBF table. We do so for convenience only.

- Always remember that SQL is the recommended file type for web and mobile applications.

Except for connecting to the database, the design process is the same for all file types.

WHO NEEDS THIS?

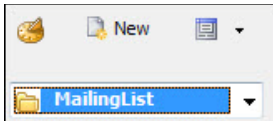
The UX component is the basis of all mobile applications and is used as a dialog in web applications. It can also be used on the desktop.

- Suitable for Desktop, Web and Mobile applications.

*. Server-side and Client-side are explained in "Understanding the Client / Server model" on page 223.

- Essential for Mobile applications.

Preparation for the lesson



In Chapter 2, you created a project entitled *MyMailingList*. We will switch to a different one for this chapter. The *Mailing List* project has copies of the components you created earlier, as well as others we will create here and in Chapter 4.



1. *Web Projects Control Panel* > *Projects*: Choose **Mailing List**.
2. Select **Web Components** to continue the lessons.

Understanding the UX Component

Currently, most mobile support is limited by speed, primarily network speed and the processing speed of the device – so the data packet size is critical to get reasonable performance. The data payload for a UX Component is much smaller than a grid. In most cases, unless you need some feature of a grid, the UX is a much better mobile solution.

Most mobile devices don't have much screen room, so the apps and pages tend to be very minimalistic. There seems to be a trend in conventional web applications toward single page applications. They are the hot new concept and even have an acronym – SPA.

The UX Component was previously known as the Dialog Component. It was completely redesigned for Version 11 and *greatly* enhanced for Alpha Anywhere (Alpha Five V12). This is important to remember when you are searching the Help files for assistance. The following apply:

- *Dialog* Component pages and videos.
 - This group explains the features, how to set-up and use.
- *UX* Component pages and videos.
 - This group focuses on Panels and List Controls, as well as other features added in v12. Requires basic understanding as described in Dialog Component videos above.

UX Components can be designed to look and behave like Grid Components, but there is a major difference between them.

- A Grid is always connected to a database.
- The UX may or may not be connected to a database.



NOTE

Personal Data
First Name: Susan
Last Name: Bush
Phone: (732) 842-3000

Security
User Name: SusanB
Password:
Confirm Password:

* Required Fields

Submit the form.

When it is not connected, it behaves like a “dialog” or can be used to display “lists.”

What is a Dialog?

A dialog is a form that asks the end user for information. For example, when you buy something on the web and fill in the name/address form – or register your name and security information as at left – you are using a dialog. The information is then evaluated for correct input (validated). If you have filled in all the entries correctly, the data is sent to the server. At that time, it is entered into the database and/or used in some other way determined by the developer.

Firstname
Lastname

Firstname	Lastname	City	State
John	Smith	Boston	MA
Henry	Rhodes	New York	NY
Allison	Berman	Los Ange...	CA

What are Lists?

Lists present data in a read-only format. The data can come from multiple data sources, such as static data, SQL queries, custom Xbasic code, custom Javascript code, etc. It can be displayed in either a columnar or free-form layout and can include images and buttons. When clicked, the buttons execute Javascript code that you have defined, either with your own code or Action Javascript, an Alpha Anywhere feature that writes the code for you.

At left is an example of a UX with two textbox fields (as you might see in a dialog), followed by a list of names and addresses. This is a very basic example that uses the “List Control,” another powerful Alpha Anywhere feature.

What’s the difference between Grid and UX components?

Understanding the difference between Grid and UX Components is critical because you need to know which to use when.

GRID COMPONENT

Always bound to a table.

- Data submission is automatic.
- When the Submit button is clicked, data is entered into the table.
- For employees and other authorized users familiar with the company, its policies and procedures.
- May require training for end user.
 - Example: Company rep enters an order.
- Feature rich, but has greater data payload. **Excellent for web applications.**

UX COMPONENT

May or not be bound to a table.

- Creates variables.*

*. See “Understanding Variables” on page 397.

- Data submission is manually determined by developer with Action Scripting and/or Xbasic.
- When the Submit button is clicked, the data in the variables may be entered into a table and/or used in another way.
- Gathers information from users inside or outside the company.
- No training necessary for end user.
 - Example: Customer places an order on-line.
- Data payload very small. **Excellent for mobile and web applications**
May replace a Grid in web applications when additional speed is required.

Viewing UX Components

With a simple preview selection, you will be able to see exactly how your applications will appear on mobile or on the web or on the desktop. Besides looking really cool, the mobile device simulator serves a much more important purpose. Smaller screens require tighter design, so it is important to be able to see and interact with the data in a real-life situation. We'll also show how you can actually view the component on your own phone or tablet.

Preparation for the lesson

The exercises in this chapter should be placed in the *MyMailing List* web project that was created in Chapter 2 (see “Creating a new Web Project” on page 27.). The completed examples are in the *Mailing List* project.

Creating a UX Component

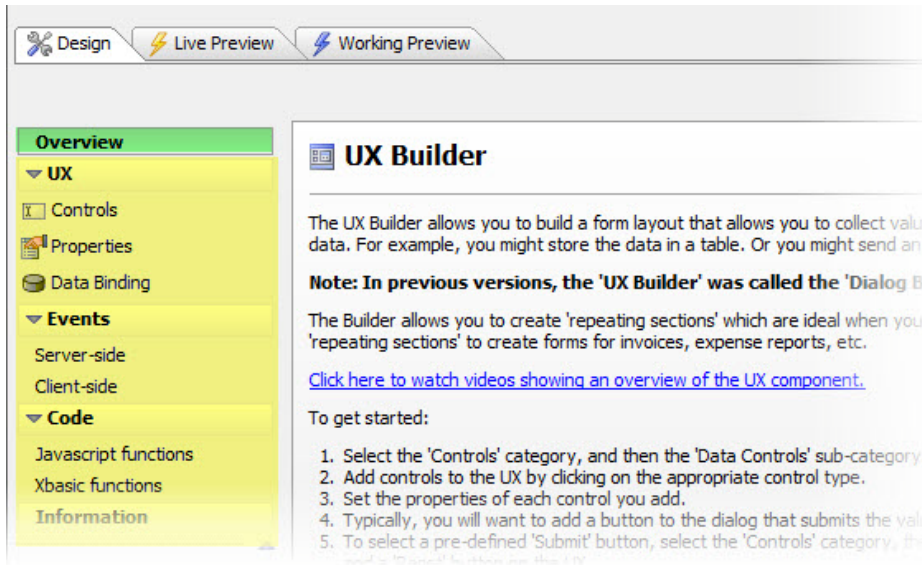
VIDEO! SEE P. 105

Overview of the UX Builder

To begin, we will take a look at the design process for UX Components.



1. *Web Projects Control Panel* > *New* > *Web Component* > *UX: Start with a blank UX Component.* (Click OK)



[Dialog: Overview](#)

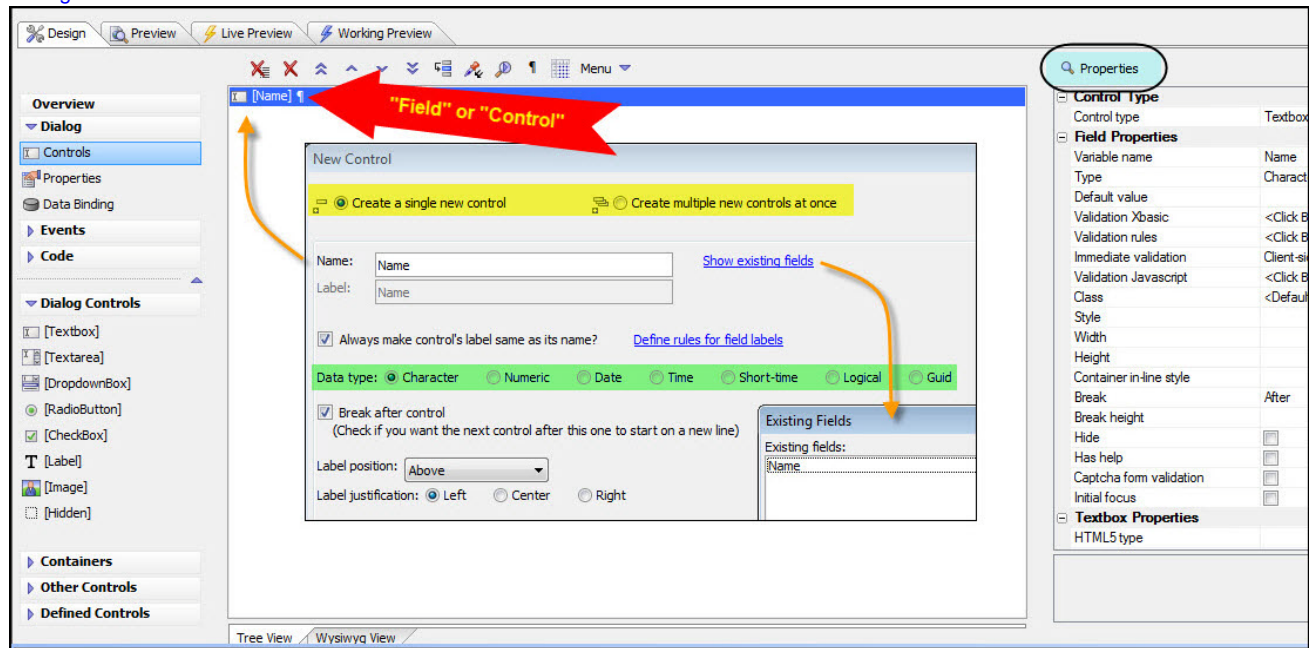
- The UX Builder appears with getting started instructions and a link to explanatory videos. You can return to this page at any time by clicking Overview in the left menu (green). These videos are also available in the Video Finder.
- The left menu navigates to the other elements of the builder (yellow).

Understanding the Menus

The menu options are extensive, permitting extreme flexibility for this component.

2. Click **UX** twice.
 - First click opens the Events, Code and Information menus.
 - Second click opens all menus.

[Dialog Title: UX Builder](#)



NAVIGATION MENU

We do not start out with *table or view* fields in the UX because it creates variables instead. To do so, it uses a series of **Controls**. The above screen shot shows the builder after one has been created.

There are five types of controls:

- Data Controls, Panels, Containers, Other Controls and Defined Controls.
- Individual controls have Properties that are shown to the right, as we saw in Grids.

The **Properties** section is for the UX itself (like Grid > Properties).

The *Properties* button (aqua above) opens the *Search Properties* dialog to quick find settings (see “Locating component controls” on page 56).

Data Binding is for mapping the controls to a table, if desired. This permits data to be entered into a table.

EVENTS & CODE

All components permit code to be added or modified.

- We prefer to work with these sections collapsed in order to open more space for the ones that follow.
- You may need to re-click UX after collapsing them in order to see the rest.

DATA CONTROLS

These controls are like the ones we saw in Grid > Field Properties > Control Types (see “Control Types” on page 34), with some additions:

- Textbox, Textarea, DropDownList, RadioButton, CheckBox, Label, Image, Hidden, Slider, Switch, Button List, List, Calendar, Tree, Map and Chart.*

PANELS

Panels are used for Mobile devices. They make the change from portrait to landscape automatic when the device is rotated. Options include:

- Panel Layout, Panel Navigator, Panel Card, Quick Panels genie.

CONTAINERS

Containers wrap controls and divide them into one or more sections:

- Tab Control, Tab Pane, Frame, Container and Free-form layout.

OTHER CONTROLS

Another selection of controls:

- Static Text, Image, Button, Hyperlink, iFrame[†], Spacer, TabStop, Embedded Object, Placeholder, FileUpload/Download.

DEFINED CONTROLS

Controls for

- Delete Record, New Record, Row Number, Update Source Grid.
- JS (JavaScript) Command Window.
- List Controls: Debugging, Navigation Buttons & Icons, Page number, Page selector, Record count, Selector, Twitter example.

*. Other controls may have been added since this was written. See each list to verify.

†. An iFrame control can be linked to a URL to show a web page.

- Record navigation buttons and icons.
- Submit-reset buttons.

Understanding UX Component design

UX design is different than Grid design.

- Since a **Grid** is bound to a table, it already has fields to which we can assign control types, such as Textbox, Radio Button, etc.
- A **UX** begins without fields, so we reverse the process, selecting the control type and then defining the **Controls** (aka fields).

TERMINOLOGY

Controls receive input from the end user. There are several types, most of which should be familiar to you: *Textbox*, *TextArea*, *DropDownBox*, *Radio Button*, *Checkbox*, etc.*

- Once created, they are listed in the UX Builder and are referred to as **Controls or Fields** (red arrow above).
- Like fields, their **Data Type** (aka **Field Type**) must be identified: *Character*, *Numeric*, *Date*, *Time*, *Short-Time*, *Logical*, *GUID*.

It is important to remember that even though the terms **Control** and **Field** are sometimes used interchangeably in UX design, the controls actually create **Variables**, a type of temporary field, not *table* fields.†

The contents of the variables can, however, be placed in table fields with a process called **Data Binding**.

Variables can also be used in other ways. In either case, Action Scripting and/or Xbasic are used to execute the **Server Side Event**.

Adding Textbox Controls

- This section continues from the previous one. A new, blank UX component should be open in Design mode.

Textbox controls are used for data entry. They can be added singly, several at a time, from a pre-defined list or from fields in a table. We will take them in turn.

Adding a single control

3. *Data Controls*: Click **Textbox** to open the **New Control** dialog (inset above).
4. Choose **Create a single new control**.
5. *Name*: Enter **Name**.

*. For more information on these controls, see “Creating Checkbox, Radio Button and Drop Down Box controls” on page 345.

†. See “Understanding Variables” on page 468.