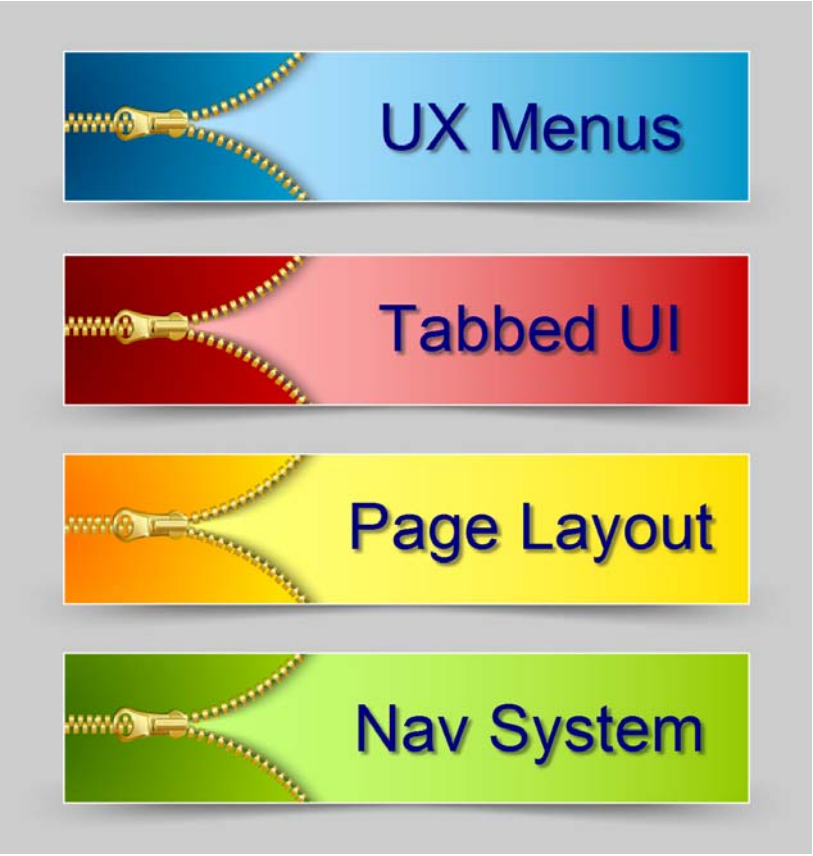
Chapter 6.

Masterful Menus



Make it easy to find…

You've worked hard to define the data, now it's time to show it off. These menu systems will gently guide your users in the right direction with friendly formats.

What you'll find here…

How the Chapter is Organized

This is where it all comes together. You will find these mobile and web menus just what you need to organize all your data.

And. It's all about presentation. A fine restaurant goes the extra mile to deliver a plate to your table that looks delicious. You are sold before you take the first bite.

And. Alpha Anywhere takes presentation seriously with elegant answers that combine your selection of components, web pages and/or reports into tidy packages. This chapter presents your recipe for success!

WEB VS. MOBILE

When it comes to menus, there is a definite split in technique between what works best for web applications and what works best for mobile where it's all about efficiency and touch.

For example, the Tabbed User Interface (Tabbed UI) is our all time favorite because it is so easy to set up and functions brilliantly, is a champ when it comes to web applications, but falls short in performance in mobile apps. Not to worry, however, because the UX has its own menu system that will give mobile users just what they expect – and then some.

With this in mind, we have split this chapter into separate sections, Mobile and Web. Here's some advice deciding which to use.

- *Mobile only developer*s: Use the UX component menu systems.
- *Web only developers:* Use the Tabbed UI, Page Layout or Navigation System menus.
- *Combination developers:* Consider two systems.
    - Web: The Tabbed UI is really easy to set up and can be used for all component types and web pages.
    - Mobile: The UX component menus take more effort to set up, but they're what you need for mobile.
    - Define who gets what with the Responsive Layout system that we described in the previous chapter (page 238).
- *Undecided*: Know you want web, but are undecided about mobile?
    - Begin with the Tabbed UI. It will handle all component types, including the UX.
    - You could also begin with the UX Menu Builder because it uses buttons, but it is not as sophisticated as the Tabbed UI.
    - The bottom line is mobile menus can always be added later.

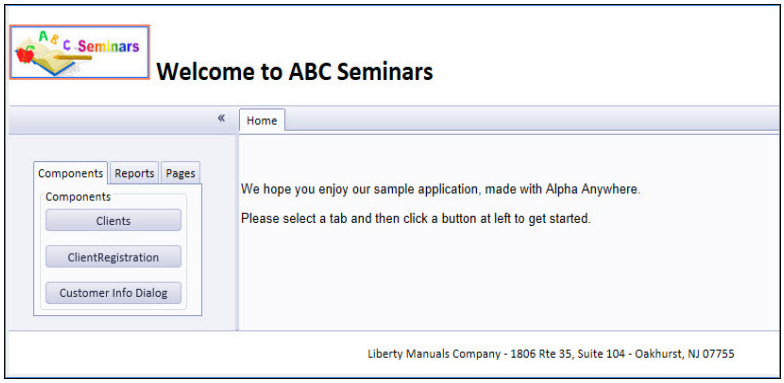
UX MENUS

Mobile. Mobile menus are designed with the UX component. We will show you how to use the Menu Builder and other quick start aids. Then we'll examine a complex system with fly out panels and some more interesting methods we

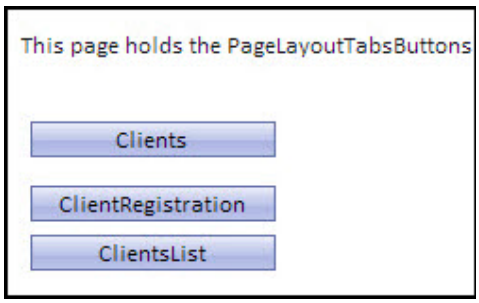
think you will find fabulous. Once you understand the principles, we know you will be off and running with your own design. (See styles below).

TABBED UI

Web: Organizes components, pages and reports with buttons and tabs. This builder even adds a Home page. Easiest & fastest.

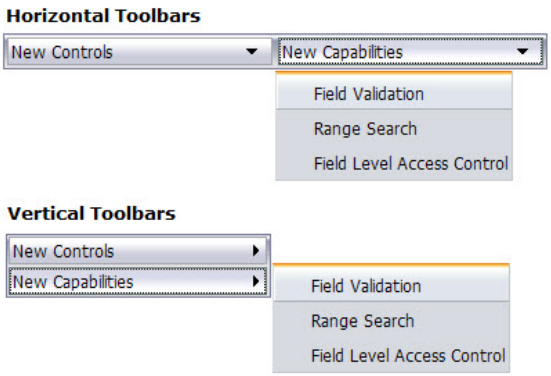


PAGE LAYOUT BUILDER



Web. Creates complex page layouts for grids, pages and reports. You can tuck them inside Tabbed UI's, too.

NAVIGATION SYSTEM



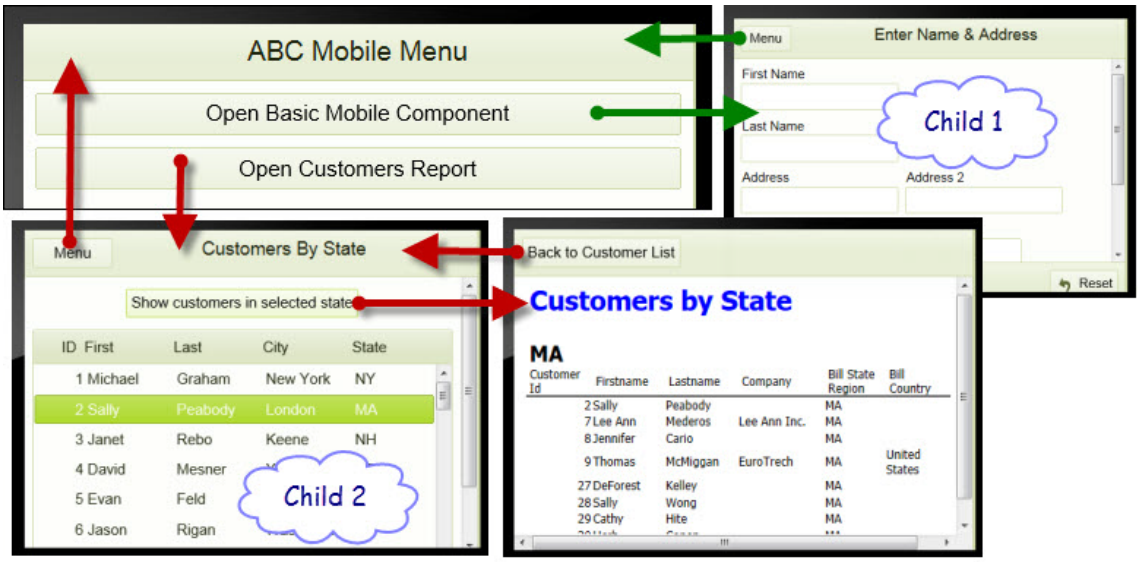
Web. This component creates a menu bar that navigates between pages. It can be set up to go across the page or down the side.

One thing is for sure – no matter where you plan to send your application, some mighty pretty packaging awaits. Yum!

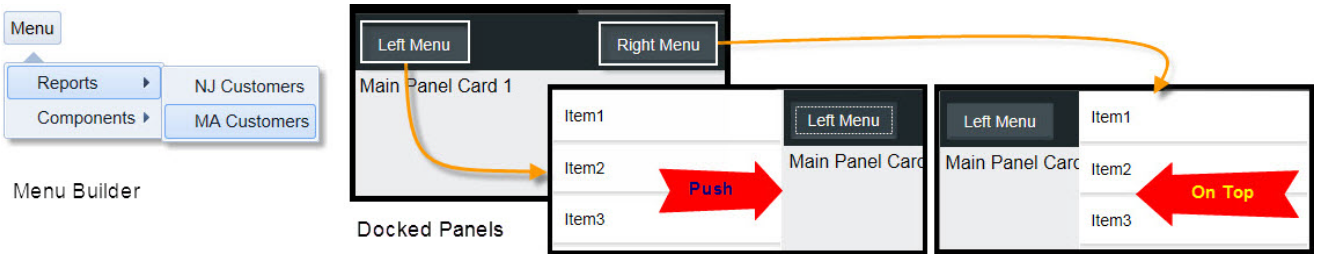
Menus for Mobile Applications:

We will begin with menus for mobile applications because we've been working with the UX up to now and it seems like a natural continuation. If you do not plan to develop mobile applications at this time, you can skip to the web menus (page 289).

In Volume 1, we showed how to create a menu that uses a combination of parent and child UX components.



This time we will work with three more types.



MENU BUILDER

Above left: The Menu Builder (left above) uses buttons to access components, reports, etc.

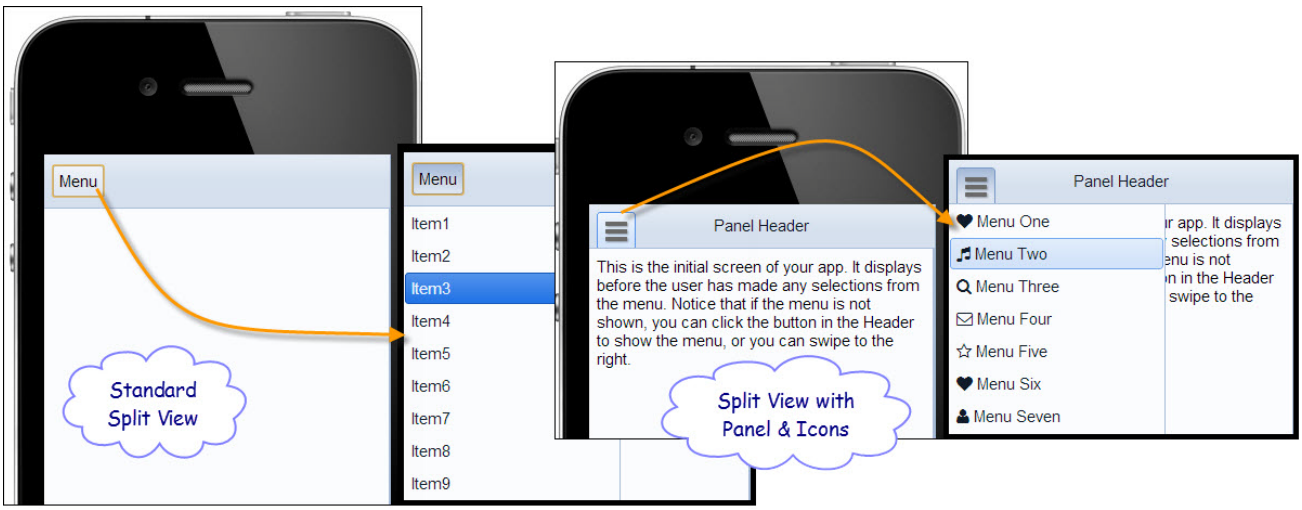
DOCKED PANELS

Above right: The opening screen displays a panel card with button(s) in the header. The menu options are "docked."

- If there is enough room to display them (like in iPad) they will appear.
- If not enough room (like in iPhone), they will be hidden until the button is pushed.
- When buttons are pushed, menus either push the panel card out of the way or display on top of the original panel.

SPLIT VIEW

Below: The Split View menus (below) also use docked panels and may or may not have panels. This exercise also shows how to add icons with the included Font-Awesome* Library. Code can also be applied for menu selections.



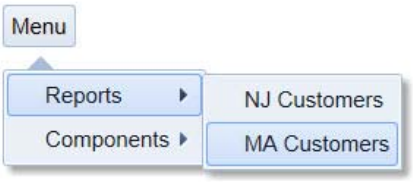
While all these menus could be used in a desktop / mouse situation, most are really designed for touch enabled devices. Here's the directory for this section:

See also:

•Volume 1, Chapter 3: Creating a UX Menu.†

A. Understanding the Menu Builder

COMPLETED COMPONENT



• The completed component for this section is MenuBuilder (Menus Web Project).

The Menu Builder is tucked inside the button *click* or *onClick* event. It can be defined with either Action Scripting, a JSON string or an Xbasic Function.

VIDEO! SEE P. 282

We will show how to do the first and let the videos explain the more advanced features.

*. "Font Awesome is fully open source and is GPL friendly. You can use it for commercial projects, open source projects, or really just about whatever you want. Attribution is no longer required as of Font Awesome 3.0 but is much appreciated: "Font Awesome by Dave Gandy - http://fontawesome.io". http://fortawesome.github.io/Font-Awesome/license/ 9.5.2014

†. Available at www.LibertyManuals.com.

These menus build on skills we've used previously, so you shouldn't have any trouble creating them. The trick is not losing your place in the nested Action Scripting actions. We'll identify the dialog titles as we go along and, as before in these multi-layered situations, we hope everyone winds up in the same place at the end!

ALSO AVAILABLE FOR GRIDS

This builder can be used with Grids, as well as UX components.

Our menu will open a couple of filtered reports, a grid and another UX. In most cases, whatever element you plan to access must be part of the current web project. To that end, we have "borrowed" a SQL report, the grid and the UX from other web projects by copying them to the Menus Web Project. The exception is DBF reports.* They just need to be in the current database, as you will see. Here's what we'll be using for this exercise:

- **Reports:**
  - **NJ Customers:** Control Panel > Reports tab > ClientsList (filtered DBF report).
  - **MA Customers:** WPCP > Menus Web Project > Reports > CustomersByState (filtered SQL report).
- **Components:**
  - **UX:** WPCP > Menus Web Project > AbsoluteLayout.
  - **Grid**: WPCP > Menus Web Project > ClientRegistration.

Creating the menu

As we said above, the Menu Builder is defined inside the *click* or *onClick* event of a button. We will open the completed component first so you can see how it works. Then we will create a new one from scratch.

1. *WPCP > Menus Web Project:* Open **MenuBuilder** in Design mode.
2. Go to **Working Preview** and click the **Menu** button.
3. Experiment with some of the menu options to **open the reports** and **components**.
4. When finished, **close** the component.
5. *WPCP > Menus Web Project:* Create a **new, blank UX component**.
6. **Save** as **MyMenuBuilder.**
7. *Properties > Style name:* Choose **MobBlue**.
8. *Controls > Other Controls*: Add a **Button** control.

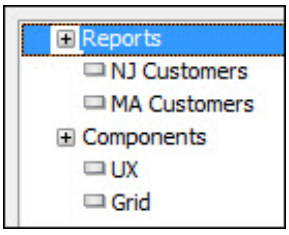
PROPERTIES: [BUTTON]

BUTTON PROPERTIES

- *Button text:* **Menu**.

*. SQL reports defined at the database level also fall into this category. In most cases, however, it is best to define reports based on SQL data at the Web Project Level.

JAVASCRIPT - (TOUCH, MOUSE, POINTER EVENTS)



- *click:* Click the button.
  - a. *Action Javascript* > **Add New Action**.
  - b. *Filter list:* Enter **Menu**. Choose: **Menus**. (Click OK)

Dialog Title: Action Javascript - Dropdown Menu

  - c. *Method or defining menu:* **Menu Builder**.
  - d. *Menu definition:* Click the button.

Dialog Title: Tree Data Genie

  - e. **Create a Tree** by adding siblings and children as at left.

Defining the "leaf" items

Here's where the Action Javascript gets tricky because there are so many layers. We'll show how to do the first and then you can complete the others. Before we start, examine the properties options available for each leaf.

- **Icon**: Choose from built-in or user-defined icons.
- **Security**: Full security options are available.
- **Show/Hide**: You can define Server-side show/hide expressions.

PROPERTIES: NJ CUSTOMERS

LEAF PROPERTIES

- *onClick:* Click the button.

Dialog Title: Edit Action

  - a. Click Create/Edit Javascript Actions.

Dialog Title: Define JavaScript Actions

  - b. Click **Add New Action**.
  - c. *Specify a unique name for the action:* Enter **NJ Cust**. (Click OK twice)
  - d. Select *NJ Cust* and click **Edit Action**.

Dialog Title: Edit unbound Event

  - e. *Action Javascript* > **Add New Action.**
  - f. *Filter list:* Enter **report**. Choose **Open a Report, Label or Letter layout.** (Click OK)

Dialog Title: Action Javascript - Open a Report, Label or Letter layout

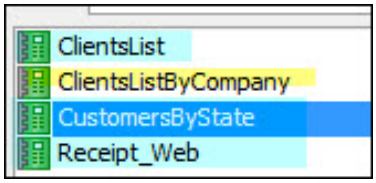
  - g. *Report name:* Click the button.

Dialog Title: Report Print Genie

  - h. Click **Select**.

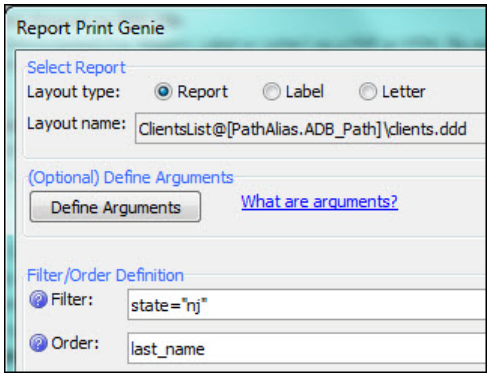
Dialog Title: Select Layout

  - i. Select **ClientsList.**



Three of the available reports are based on DBF data in the current database (aqua). The fourth is a SQL table and is part of the current web project (yellow). We will use a DBF table here and the SQL one for MA customers.

  - j. Click OK.



Dialog Title: Report Print Genie

k. *Filter:* Click **Build** and define as follows:

- **state** is equal to **nj** (Click OK)

l. *Order:* Click **Build** and define as follows:

- *Sort by:* **last_name**. (Click OK twice)

Dialog Title: Action Javascript - Open a Report, Label or Letter layout

m. *Initial View*: **PDF**.

- HTML option does not work with DBF reports.*

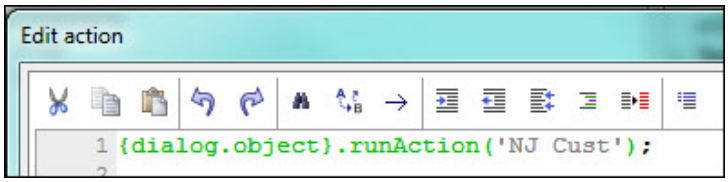
n. Click OK.

o. *Comment*: **Open ClientList report, State = NJ**. (Click OK)

p. Save the Action Javascript.

Dialog Title: Define Javascript Actions

q. Click OK.



Dialog Title: Edit action

r. Click Run a Javascript Action.

Dialog Title: Select Javascript Action to Run

s. Choose **NJ Cust**. (Click OK)

Dialog Title: Edit action

t. Click OK.

Dialog Title: Tree Data Genie

- **Define** other leaves as follows:

u. MA Customers:

- *Action name:* **MA Cust**.
- Use Action Javascript to open a report as above.
  - *Report name:* **CustomersByState**.
  - *Filter:* **Bill_State_Region = 'MA'**
  - *Order:* **Lastname**.

v. UX:

- *Action name:* **Open UX Component**.
- Use Action Javascript to open a UX component.
  - *Component name:* **AbsoluteLayout**.

w. Grid:

- *Action name:* **Open Grid Component**.
- Use Action Javascript to open a Grid component.
  - *Component name:* **ClientRegistration**.

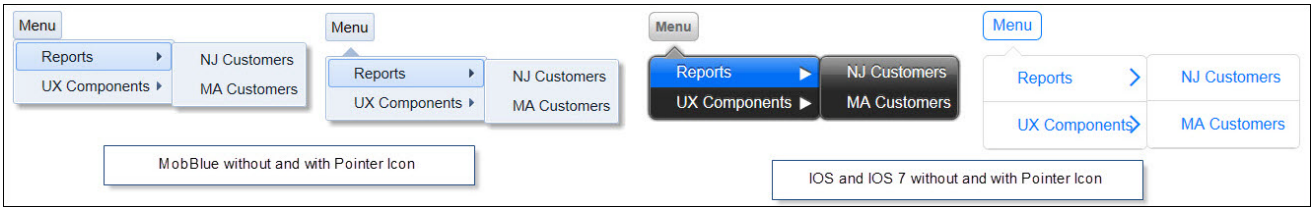
x. When finished, Click **OK** to close the Tree Data Genie dialog.

*. If you do choose HTML for a DBF report, at runtime you will get a somewhat mysterious error message saying that the report must be Layout Table style. Making this correction does not clear the error message, but changing the initial view to PDF does.

POINTER ICON

The Menu Builder has two styling options, with and without pointer icon(s). The effect is dependent on the style of the UX itself.

- If the style is MobBlue or a similar style, the ***option*** will add a pointer to the button (below left).
- If you are using either IOS or IOS7 style, the pointer will appear ***regardless*** of the Menu Builder setting.



Dialog Title: Action Javascript - Dropdown Menu

y. *Menu Properties > Has pointer icon:* **Yes**.

- See notes above. IOS and IOS7 *always* have pointers.

z. *Pointer position:* **Auto**.

- Other options are Top, Bottom, Left and Right.

aa. *Menu Animation > Animation:* **Yes**.

- Full selections of animation styles, speeds, etc. is available.

ab. Click OK.

ac. *Comment:* **Define Menu Builder**.

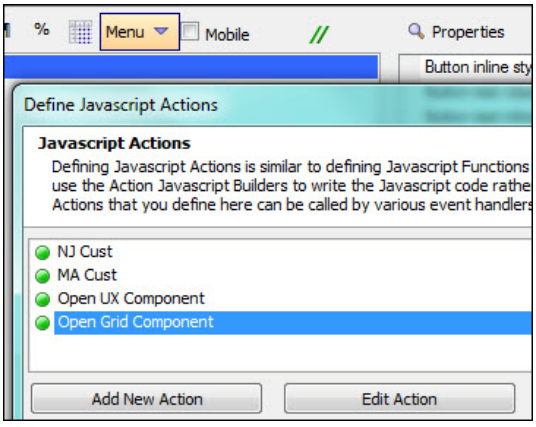
ad. **Save** the Action Javascript.

Hopefully, you will now be at the UX builder.

9. **Save** the component and go to **Working Preview** and examine the results of all your hard work!

MOBILE DEVICES

As always, if you plan to use this for mobile devices, add a Panel Card in the usual manner.



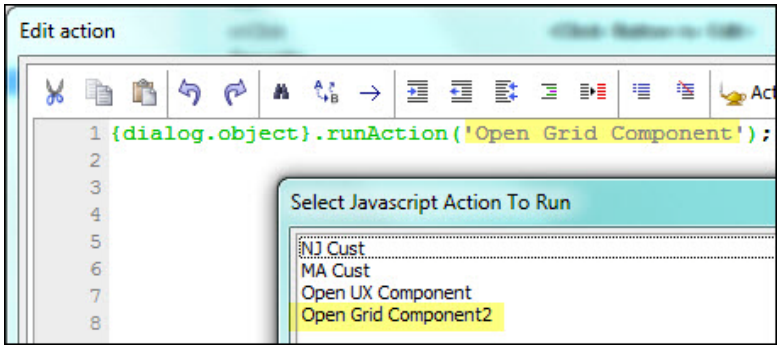
ACTION JAVASCRIPT EDITING SHORTCUT

This one place where the Javascript Actions shortcut can really come in handy if you need to make some edits. We've seen it before, but it is worth a review here.

10. Return to **Design** mode.

11. Click **Menu** (above controls list) and choose **Javascript Actions Editor**.

The existing actions are displayed and can be edited, deleted or renamed from here.



CAUTION

Keep in mind that if you **rename** here, it will affect connections made elsewhere. For example, renaming one of these connections means the Javascript in the Tree Data Genie will no longer be valid unless you go in and change the code to the renamed action.

This editor is ideal, however for editing the action itself.

12. **Close** the component.

GO TO ALPHA VIDEO
UX_V12-19, 20, 21

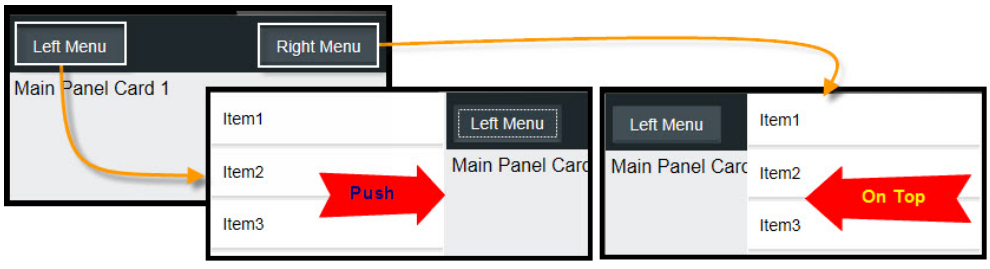
- Videos > Filter: Enter **Menu**. Choose:
  - UX_V12--19. Menus - Displaying pop-up menus when the user clicks a button.
  - UX_V12--20. Populating Menus using a JSON String or an Xbasic Function. Advanced developers.
  - UX_V12--21. Menus - Advanced features - menus can include forms, buttons, etc. Advanced developers.

B. Building a Menu with Docked Panels

COMPLETED COMPONENT

- The completed components for this section are jQueryDemo, jQueryDemo-Simple, RichNavigation (Menus Web Project).

This menu style uses a list that comes into sight with the click of a button or swipe action and moves out of the way when the selection has been made. It is especially good for small devices like phones because the menus do not take up any screen real estate when they are not in use. This type of menu employs *Docked Panels*.*



Docked Panels can have two different behaviors. They can either push the Panel Card out of the way or they can move over top of it. This exercise will demonstrate both.

HOW IT WORKS

The initial view shows only the main Panel Card. When the user clicks a button in the header, the docked Menu moves into view. He/she then selects an item on the list and is taken to the specified Panel Card.

*. See “Understanding Docking” on page 151.