

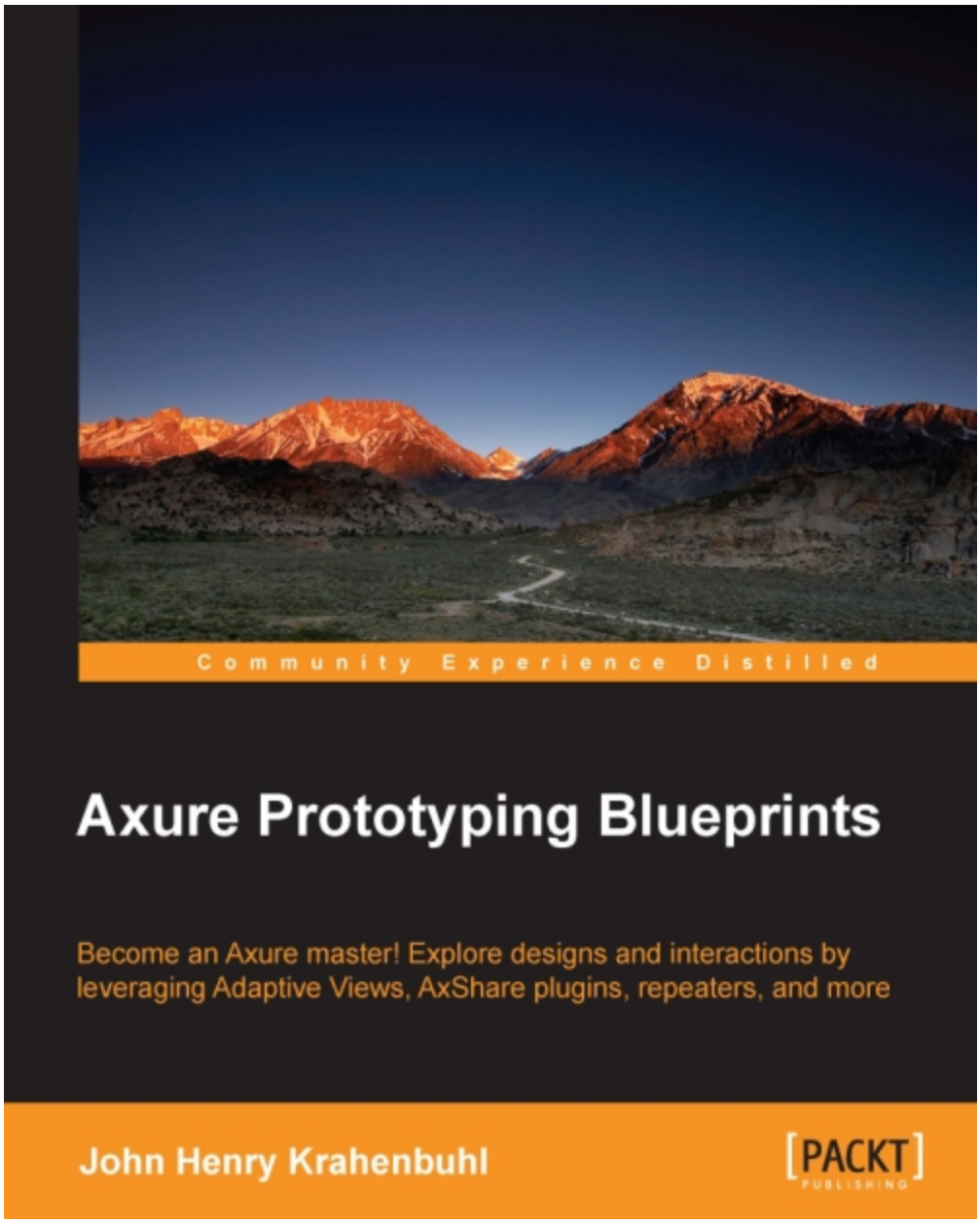
Community Experience Distilled

Axure Prototyping Blueprints

Become an Axure master! Explore designs and interactions by leveraging Adaptive Views, AxShare plugins, repeaters, and more

John Henry Krahenbuhl

[PACKT] PUBLISHING



Community Experience Distilled

Axure Prototyping Blueprints

Become an Axure master! Explore designs and interactions by leveraging Adaptive Views, AxShare plugins, repeaters, and more

John Henry Krahenbuhl

[PACKT] PUBLISHING

Table of Contents

Axure Prototyping Blueprints
Credits
About the Author
About the Reviewers
www.PacktPub.com
    Support files, eBooks, discount offers, and more
        Why subscribe?
        Free access for Packt account holders
Preface
    What this book covers
    What you need for this book
    Who this book is for
    Conventions
    Reader feedback
    Customer support
        Downloading the example code
        Errata
        Piracy
        Questions
1. Laying the Foundation
    The environment and interface
        Pages
            The Sitemap pane
            Creating a child page
        The design area
            Opening pages
            Showing the grid
            Adding page guides
        Panes
            The widgets pane and libraries
            Masters
            The Page Properties pane
            The Widget Interaction and Notes pane

The Widget Properties and Style pane
The Widget Manager pane
Adaptive Views
Planning for adaptive
Managing Adaptive Views
Summary
2. Building a Community Site
Reviewing our project
Background information for our client
Architecting our community site
Inspecting our sitemap
Reviewing the forum and news flow
Identifying and building our masters
Creating global variables
Adding masters to the Masters pane
Designing our Header master
Placing widgets on the Header master
Building interactions for our Header master
Defining interaction for CommunityLogo
Enabling interactions for HzMenu
Illuminating interactions for EmailTextField
Activating the OnMouseEnter event
Facilitating the OnMouseOut event
Constructing interactions for the PasswordTextField
Activating the OnMouseEnter event
Facilitating the OnMouseOut Event
Constructing our BreadCrumb master
Formatting the Repeater widget and placing widgets on the BreadCrumb master
Updating the Repeater item's OnItemLoad event for the BreadCrumb repeater
Defining the OnItemLoad event actions
Assembling our Secondary_Page_Body master
Placing widgets on the Secondary_Page_Body master
Manufacturing our Informational_Page_Body master
Placing widgets on the Informational_Page_Body master

Fabricating our Forum master
Formatting the Repeater and placing widgets on the Forum master
Updating the Repeater item's OnItemLoad event for the Forum repeater
Realizing OnItemLoad event actions
Defining the UpdateTopic case
Defining the UpdateCategory case
Building interactions for the Forum master
Enabling the OnClick interactions for the Category Hot Spot
Activating the OnClick interactions for the Topic Hot Spot
Forming our Footer master
Placing widgets on the Footer master
Building interactions for our Footer master
Defining interactions for our Label widgets
Finalizing page design for our community site
Placing widgets and masters on the Home page
Designing the page body
Placing widgets and interactions for the dynamic carousel
Building interactions for the carousel dynamic panel
Enabling interactions for the CheckForClick Hot Spot
Defining the OnClick interactions for the Carousel dynamic panel
Activating the Previous and Next buttons
Placing widgets and adding interactions for calls to action
Adding a social media feed
Adding masters and defining actions for the home page
Completing the design and interactions for the Journey, Topic Detail, Charter, and About Us pages
Defining interactions for the Journey, Topic Detail, Charter, and About Us pages
Enabling OnPageLoad actions for the Journey page

Activating OnPageLoad actions for the Topic Detail page
Facilitating OnPageLoad actions for the Charter page
Constructing OnPageLoad actions for the About Us page
Manufacturing design and interactions for the Forum page
Defining OnPageLoad actions for the Forum page
Fabricating design and interactions for the Topic List page
Defining OnPageLoad actions for the Topic List page
Creating the Initialize case for the OnPageLoad event
Adding the SetAllTopicFilter case for the OnPageLoad event
Completing the design and interactions for the Terms, Privacy, and Help pages
Creating the Initialize case for the OnPageLoad event for the Terms page
Creating the Initialize case for the OnPageLoad event for the Privacy page
Creating the Initialize case for the OnPageLoad event for the Help page
Summary
3. Constructing a Blog
Reviewing our project
Background information for our blog
Planning our blog
Inspecting our sitemap
Reviewing the registration and post flows
Updating and building our masters
Creating additional global variables
Adding masters to the Masters pane
Enhancing our Header master
Refining the interactions for our Header master
Defining interactions for the LoginButton
Validating the user's email and password
Creating the ShowErrorMessage case

Enabling interaction for the NewUserLink
Activating interactions for the ForgotLink
Facilitating interactions for the LogOutButton
Constructing interactions for the RegistrationLightBoxDP
Updating Variables and hiding the RegistrationLightBoxDP
Creating the ShowLoginState case
Refreshing data for our Forum repeater
Assembling our Post Commentary master
Defining the OnItemLoad event actions
Manufacturing a NewPost master
Repurposing pages in our sitemap
Remodeling our home page
Invigorating our remaining pages
Enabling commentary and new post creation
Facilitating commentary creation
Allowing new post creation to authorized users
Summary
4. Leveraging Social Content Aggregation
Developing an aggregator
Building our Social Content master
Adding Facebook and Twitter to our feed
Pulling Instagram content
Including a Pinterest board
Summary
5. Showcasing Portfolio Content
Designing our parallax site
Placing page anchors
Building the background dynamic panel
Creating our sectional content
Adding additional portfolio interactions
Updating page interactions
Defining the OnPageLoad event
Enabling OnWindowScroll interactions
Defining the WindowScrolled case
Completing the remaining OnWindowScroll cases
Defining the WindowScroll Y < 400 case

Enabling the WindowScroll Y > 400 & Y < 800 case
Creating the WindowScroll Y > 800 case
Summary
6. Creating an eBrochure
Creating our eBrochure page
Designing content with Axure for our eBrochure
Designing our cover page
Designing page1
Designing page2
Designing page3 using the left-hand side of our panoramic image
Designing page4 using the right-hand side of our panoramic image
Designing our back page
Converting pages into images
Creating a page flipper AxShare plugin
Publishing to AxShare
Summary
7. Creating an eMagazine
Designing an eMagazine
Updating the sitemap and configuring Adaptive Views
Creating global variables
Adding masters to the Masters pane
Designing the Header master
Designing the Footer master
Designing the eMagazine home page
Designing the Topic Overview page
Defining the OnPageLoad interaction
Defining OnClick interactions
Designing the Article Detail page
Defining the OnPageLoad interaction
Defining the OnClick interactions
Summary
8. Creating a Photo Contest
Designing a photo contest
Updating the sitemap and configuring Adaptive Views
Creating global variables

- Adding masters to the Masters pane
- Designing the Header master
- Designing the Footer master
- Designing the Photo Contest home page
- Creating the Enter Now flow
  - Designing the Enter page
  - Designing the Upload Photo page
- Designing the Entry Confirmation page
- Designing the Gallery page
  - Designing the PhotoGallery repeater
    - Updating the PhotoGallery repeater for the 768 and 320 views
  - Defining PhotoGallery repeater interactions
  - Defining sorting and pagination interactions
    - Enabling sorting for SortDroplist
    - Enabling pagination controls
- Designing the View Entry page
- Designing the Prizes page
- Designing the Rules page

Summary

9. Creating an E-commerce Shopping Cart

Designing an e-commerce shopping cart

- Updating the sitemap and configuring Adaptive Views
- Creating global variables
- Adding masters to the Masters pane
- Designing the Header master
- Designing the Footer master
- Designing the Shopping Cart page
  - Designing the Cart repeater
    - Updating the Cart repeater for the 768 and 320 views
  - Defining the Cart repeater interactions
    - Creating the Quantity widget's OnSelectionChange case
    - Creating the DeleteButton widget's OnClick cases
    - Creating cases for the Cart repeater OnItemLoad event
  - Creating the sticky Proceed to checkout button
  - Enabling the OnWindowScroll page interaction

Summary

Index

Axure Prototyping Blueprints

Axure Prototyping Blueprints

Copyright © 2015 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: July 2015

Production reference: 1280715

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham B3 2PB, UK.

ISBN 978-1-78355-197-2

www.packtpub.com

Credits

Author

John Henry Krahenbuhl

Reviewers

Ryan J. Flynn

Jan Tomáš

Commissioning Editor

Usha Iyer

Acquisition Editor

Subho Gupta

Content Development Editor

Athira Laji

Technical Editor

Pankaj Kadam

Copy Editors

Puja Lalwani

Kausambhi Majumdar

Laxmi Subramanian

Project Coordinator

Harshal Ved

Proofreader

Safis Editing

Indexer

Rekha Nair

Production Coordinator

Manu Joseph

Cover Work

Manu Joseph

About the Author

John Henry Krahenbuhl has over 20 years of experience in architecting practical, cost-effective, and innovative solutions. Being a creative thinker and having an entrepreneurial spirit has enabled him to perform in the role of lead or co-inventor on at least seven granted patents as well as numerous pending utility patent applications. He is a multifaceted and collaborative management professional, who is highly skilled at managing products through their entire life cycle from design to obsolescence, including specification and use case definitions, schematic and PCB layouts, production software implementation, and hardware implementation.

He's a passionate, resourceful leader who demands and delivers excellence in design and user experience. To his credit, John is also the author of *Axure RP Prototyping Cookbook* as well as *Learning Axure RP Interactive Prototypes*, both published by Packt Publishing.

My gratitude and thanks go to my family.

To my children: Matt, Jason, Lauryn, and Henry. I thank you for your understanding and patience, as I spent countless hours of our family time in writing and prototyping.

To Melissa, my wife and the love of my life, thank you for always being supportive, believing in me, and keeping me focused and on track. Every day, you fill my heart with joy and I am so blessed to share my life with you!

About the Reviewers

Ryan J. Flynn is a designer and an artist in New York. A lover of information and simplicity, he has found himself crafting solutions and interfaces for retail, wayfinding, wearables, financial, marketing, pharmaceuticals, legal, and everything in between. When he isn't trying to uncomplicate things, he spends his free time seeking out the tastiest parts of Brooklyn.

Jan Tomáš is the owner and designer at the UX agency CIRCUS DESIGN (www.circusdesign.cz/en), based in the Czech Republic.

His specialization is with user research and prototyping. He uses Axure RP on a daily basis to prototype web and mobile applications to communicate his designs with developers, managers, and other stakeholders. He has gained experience from large international projects. Jan is also a Windows UX design specialist and is certified by Microsoft. For more information about Jan, visit www.jantomas.cz.

www.PacktPub.com

Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at <service@packtpub.com> for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



https://www2.packtpub.com/books/subscription/packtlib

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

Preface

Axure continues to garner rave reviews from the design community and is one of the leading tools for rapid prototyping that is in use today. With Axure, one can easily create wireframes and clickable prototypes. Using AxShare (Axure's cloud-based prototype-hosting service) and the new AxShare mobile app (http://www.axure.com/learn/axshare/app), prototypes can be easily shared with collaborators and clients alike.

These are just a few reasons for Axure's continued popularity. When Axure 7 was released, the Axure community was presented with a new framework. The improved framework also supported adaptive prototyping.

Additional enhancements to Axure (for example, the Repeater widget) and AxShare (for example, the AxShare plugins) provide a new level of prototyping excellence. With Axure RP 8 and Axure RP 8 Pro, the bar has once again been raised.

Axure Prototyping Blueprints will present to the novice, as well as experienced users, common design patterns for exploration and continued learning. Armed with these new design patterns and methods, readers can complete projects faster and with increasing levels of interactivity.

This book has been a labor of love and it is my third book on Axure and prototyping. I wish that *Axure Prototyping Blueprints* empowers you on your journey to Axure mastery. My other two books, *Axure RP Prototyping Cookbook* and *Learning Axure RP Interactive Prototypes*, both by Packt Publishing, complete the Axure trilogy. I hope my books will increase your knowledge and become reference materials for you in the future. You should know that, in addition, the Axure community (http://www.axure.com/community) is always ready to provide additional insights and assistance, if the need arises.

What this book covers

Chapter 1, *Laying the Foundation*, provides an introduction for novice users and a refresher for more experienced users to fundamental Axure terminology, interfaces, and concepts. A high-level introduction to the Adaptive View feature, released in Axure 7, will also be provided.

Chapter 2, *Building a Community Site*, presents interactions for a header, an interactive carousel, and includes a right column with live feeds from social media channels such as Facebook and Twitter.

Chapter 3, *Constructing a Blog*, provides the user an opportunity to explore design patterns and interactions for a typical blog (for example, as often implemented in WordPress).

Chapter 4, *Leveraging Social Content Aggregation*, provides a method to pull into Axure the live feeds from some of the most popular social media channels (for example, Facebook, Twitter, Instagram, and Pinterest).

Chapter 5, *Showcasing Portfolio Content*, simulates a sample portfolio site that includes parallax scrolling.

Chapter 6, *Creating an eBrochure*, teaches how to expand Axure's core set of interactions by using an AxShare plugin with JavaScript.

Chapter 7, *Creating an eMagazine*, teaches how to create an eMagazine optimized for desktop, tablet, and mobile viewports with the new Adaptive View feature.

Chapter 8, *Creating a Photo Contest*, teaches how to build a photo contest with registration and a photo gallery, and view entry pages optimized for desktop, tablet, and mobile viewports with the Repeater widget and the Adaptive View feature.

Chapter 9, *Creating an E-commerce Shopping Cart*, teaches how to architect a shopping cart for an e-commerce site optimized for desktop, tablet, and mobile viewports with the Repeater widget and the Adaptive View feature. This chapter will also enable you to do common interactions such as removing and updating the number of items in the cart.

What you need for this book

You will need Axure 7, Axure RP 7 or later, an Internet connection, and a desire to learn interactive prototyping. If you do not currently have Axure, please visit http://www.axure.com to download a free trial version.

Who this book is for

You are a user experience professional, designer, information architect, or business analyst who wants to explore common design patterns and enhance your interactive prototyping skills with Axure. You have some experience in creating wireframes, as well as an interest in interaction design, and want to take your interactive prototypes to the next level.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "The URL will typically be shown on the third line starting with `src=`."

New terms and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "The **Sitemap** pane provides a hierarchical overview of the pages in your design."

> ## Note
>
> Warnings or important notes appear in a box like this.

> ## Tip
>
> Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail <feedback@packtpub.com>, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files from your account at http://www.packtpub.com for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit http://www.packtpub.com/support and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting http://www.packtpub.com/submit-errata, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to https://www.packtpub.com/books/content/support and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at <copyright@packtpub.com> with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

Questions

If you have a problem with any aspect of this book, you can contact us at <questions@packtpub.com>, and we will do our best to address the problem.

Chapter 1. Laying the Foundation

As with any structure, the foundation is critical for stability. Having a solid understanding of Axure's environment and interface provides us with a solid foundation. Once you are familiar with specific aspects of the environment, you will be able to rapidly create interactive prototypes. The interface comprises the main menu, a toolbar, a sitemap, the design area, and several panes.

In this chapter, you will learn about:

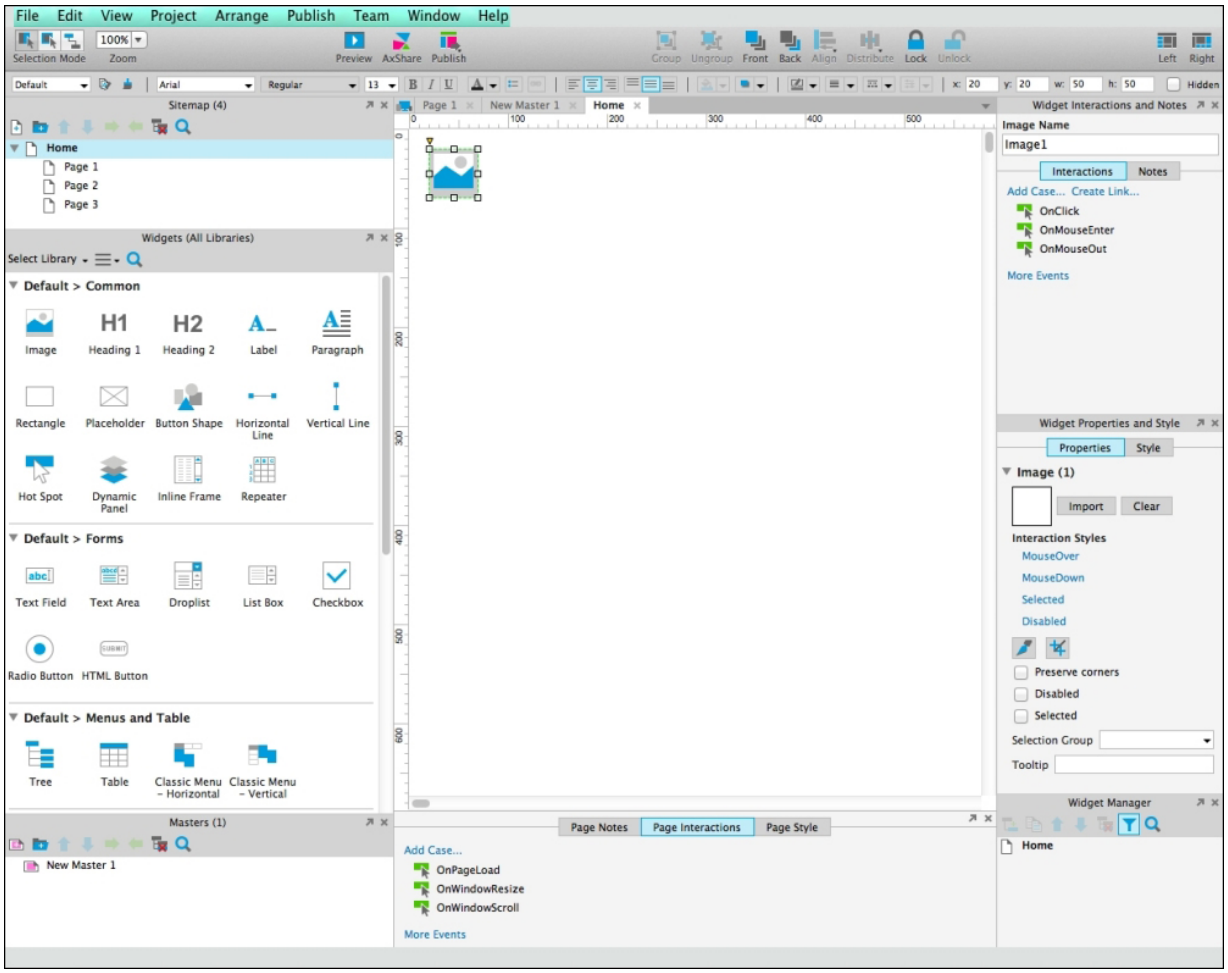
- The environment and interface
- Adaptive Views

The environment and interface

Axure's environment has a similar feel to other desktop applications. We will now explore the environment and interface.

> **Note**
>
> Please open a blank Axure document so we can familiarize ourselves with the tool. In future, this chapter can serve as a quick reference for Axure interface elements. More advanced users may want to skim this section as a refresher.

When you open a blank Axure document, the interface looks like this:



Note

The preceding screenshot depicts the Mac version of Axure RP. The PC version may differ slightly in appearance.

Axure's main interface is organized into sections: the main menu, the toolbar, the design area, and the surrounding panes. First, you will explore the interface as follows:

At the top is the **main menu**, which contains the following menus:

- **File**: This provides us with options such as **New** for creating new RP files, **Open**, **Save As**, and **Import from RP File**, and

also allows us to print, export, backup, and recover.
- **Edit**: This menu facilitates options such as **Cut**, **Copy**, **Paste**, **Find**, and **Replace**.
- **View**: This menu has options such as **Panes**, **Toolbars**, **Reset View**, **Masks** and show options.
- **Project**: This menu covers the options for widget and page (for example, style editors and note fields), Adaptive Views, global variables, and project settings.
- **Arrange**: This menu has options such as **Group**, **Ungroup**, **Bring to Front**, **Send to Back**, and alignment (for example, left, right, center, and so on). It also covers options for grids and guides.
- **Publish**: This menu facilitates us with the new **Preview** option that enables quickly viewing your prototype, the **Generate HTML Files** option, and the option to generate specifications. It also has the **More Generator and Configuration** option.
- **Team**: Team projects allow you to share and work on single projects in a team environment. Items in this menu are **Generate Team Project from Current File**, **Get and Open Team Project**, and **Browse Team Project History**.
- **Window**: This covers the **Minimize** and **Zoom** options.
- **Help**: This menu will have options to search, access Axure forums, manage license keys, and check for updates.

Near the top of the RP interface is a **toolbar**. The toolbar comprises two sections stacked vertically:

- The top toolbar section contains the following options:
    - **Selection mode**: Intersected Mode, Contained Mode, and Connector Mode
    - **Zoom**: A dropdown to set the default zoom level
    - **Publishing**: Preview your prototype, AxShare (for example, publish your prototype to Axure's cloud-based sharing service), and options from the **Publish** menu
    - **Arrangement of widgets**: **Group**, **Ungroup**, **Front**, **Back**, **Align**, **Distribute**, and **Lock** and **Unlock** widget positions

    - **Interface layout**: **Left** or **Right**. Click on this to toggle (show or hide) left and right panes
- The bottom toolbar section contains the following options:

    - Dropdown for selected widget style
    - Widget style editor
    - Format painter
    - Font formatting options (for example, font family, typeface, size, bold, italics, underline, color, bulleted list, and insert text link)
    - Text alignment (for example, left, center, right, top, middle, and bottom)
    - Fill color
    - Outer shadow
    - Line options (for example, color, width, pattern and arrow style)
    - Widget location and visibility (for example, coordinates (x,y), **w:** for width, **h:** for height, and hidden)

In the center of the interface is the **design area**. The design area is where you open pages, and drag and drop widgets to build your interface. The design area is surrounded to the left, right, and bottom by panes. The panes are organized as follows:

- The left-side column contains the following panes:

    - **Sitemap**: The **Sitemap** pane provides a hierarchical overview of the pages in your design.
    - **Widgets**: The **Widgets** pane shows widget libraries that enable us to rapidly create wireframes and flow diagrams.
    - **Masters**: The **Masters** pane organizes templates that can be reused throughout your design. Common uses for masters are page elements such as headers and footers.

- Under the design area, the bottom-center column contains the page properties pane, which has the following three tabs:

    - **Page Notes**

    - **Page Interactions**
    - **Page Style**
- The right-side column contains the following panes:

    - **Widget Interactions and Notes**: The **Widget Interactions and Notes** pane has two tabs: **Interactions** and **Notes** (in the Axure RP Pro version)
    - **Widget Properties and Style**: The **Widget Properties and Style** pane has two tabs: **Properties** and **Style**
    - **Widget Manager**: The **Widget Manager** pane enumerates widget states (for example, dynamic panel states, and so on)

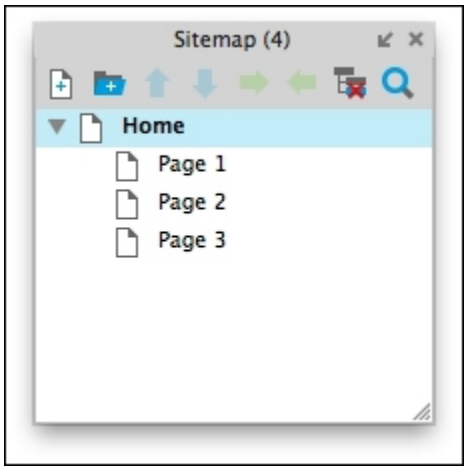
> ### Note
>
> Axure is available in two versions: Standard and Pro. The Pro version includes additional functionality such as documentation and team projects. Learn more at http://www.axure.com/compare.

Pages

Traditionally, a sitemap provides a top-down view of the pages for a website. In Axure, the **Sitemap** pane can comprise pages and folders organized in a hierarchical fashion. Just under the toolbar, you will find the **Sitemap** pane.

The Sitemap pane

When you open a blank RP document, you will see four pages in the **Sitemap** pane. The pages are labeled **Home**, **Page 1**, **Page 2**, and **Page 3**, and are organized as follows:



Note

In the **Sitemap** pane, you will notice **(4)** next to **Sitemap**. This number indicates the total number of pages currently in the **Sitemap** pane.

At the top of the **Sitemap** pane, you will see a row of eight icons as follows:

- Add Page
- Add Folder
- Move Up
- Move Down
- Indent
- Outdent
- Delete
- Search

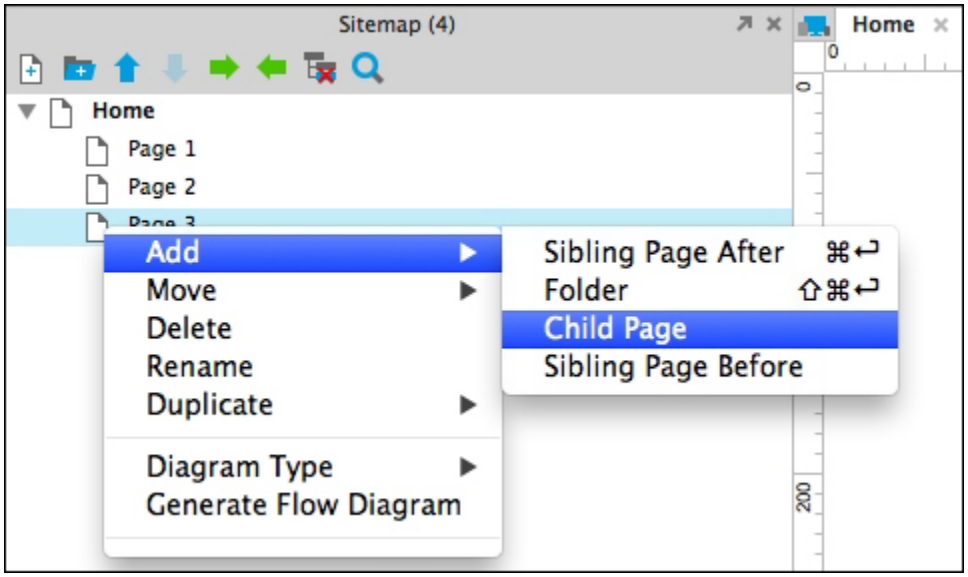
Creating a child page

Icons of the pages highest in the hierarchy are left justified. When a page becomes a parent page by having a child page association, a grey arrow is shown to the left of the page icon. The gray arrow

enables you to collapse or view pages in that branch of the hierarchy.

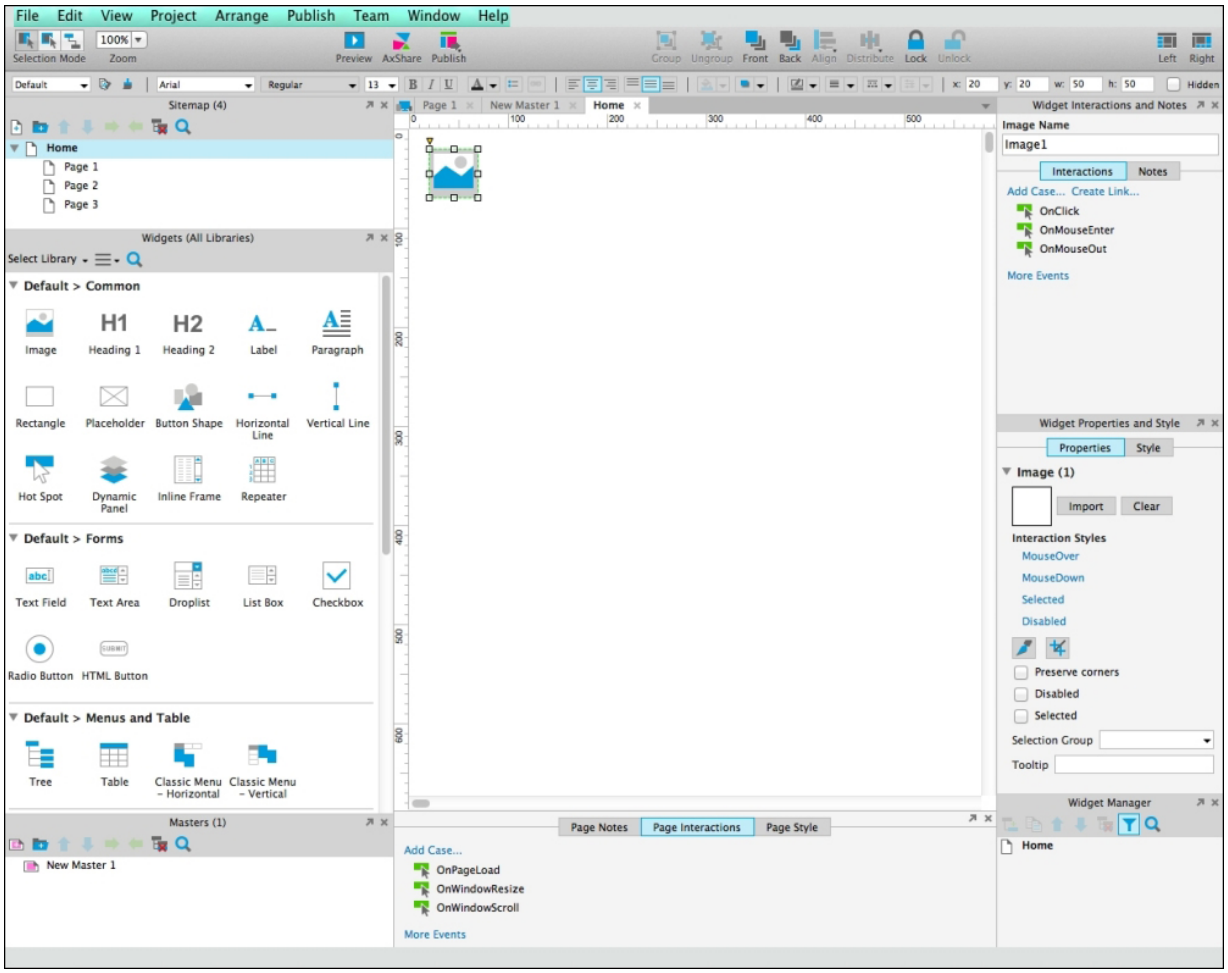
There are several ways to create a child page:

- You can create a child page by clicking on the page in the **Sitemap** pane, clicking on the indent icon, and then clicking on Move Up or Move Down icon to move the page.
- Another method is to click and drag the page to the right, and drag up or down if you need to move the page. You will see a blue box appear around the page that will become the parent page.
- You can also create child pages by right-clicking on the parent page, moving the cursor over **Add**, and then clicking on **Child Page**, as shown in the following screenshot:



The design area

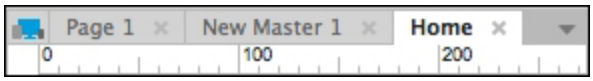
The **Design area** (that is, the blank canvas) is in the center of the interface, as shown in the following screenshot:



Opening pages

Pages can be opened in the design area by double-clicking on the page name or icon in the **Sitemap** pane. Masters are also opened in the design area by double-clicking on the master name or icon in the **Masters** pane (bottom-left). When a page or master is opened in the design area, a tab will be added to the top of the design area with the name of the page or master (for example, double-clicking on the **Home** page in the **Sitemap** pane will display that page in the design area in a tab named **Home**.)

When you open more than one page or master, the tabs will remain available in the design area until you choose to close the tab. Inactive tabs will be colored gray and the active tab will be white as shown:

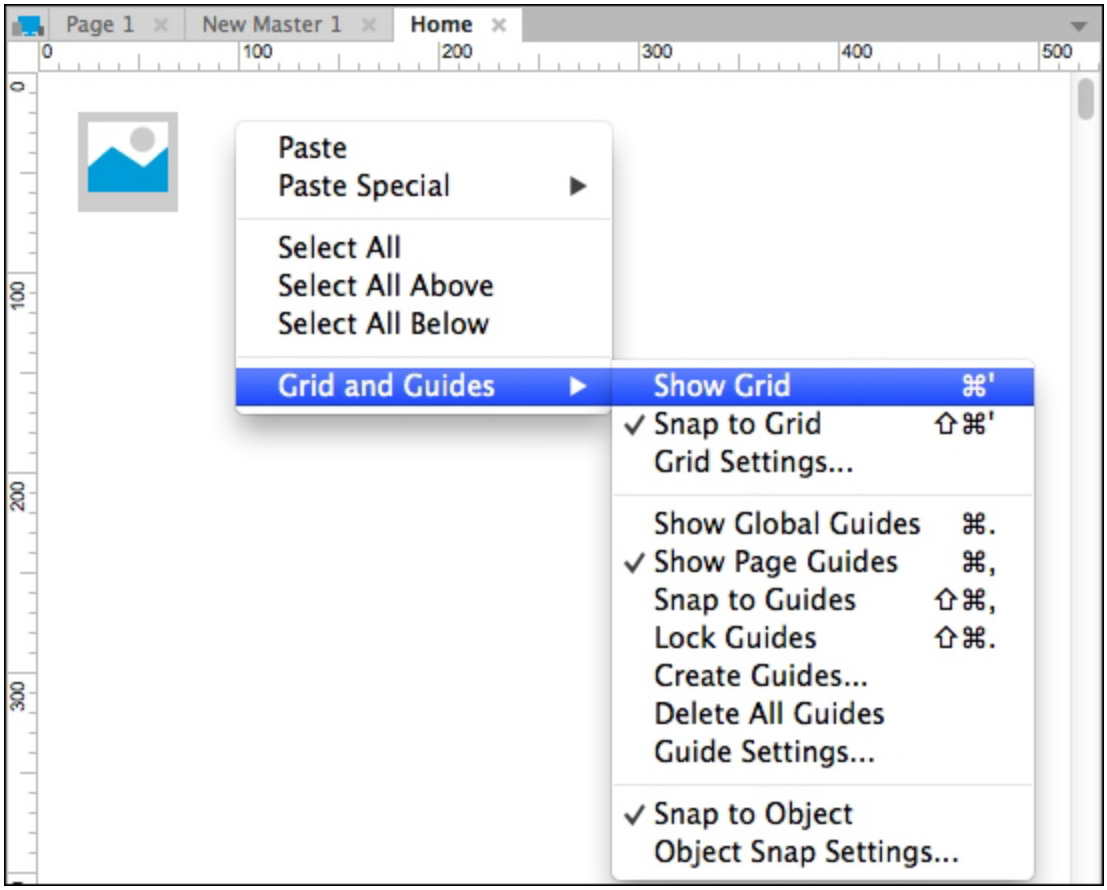


Note

To close a given tab, click the **x** next to the tab name. You can also manage tabs by clicking on the down arrow to the right of the tab names.

Showing the grid

When placing widgets in the design area, you may prefer having a grid to use as a visual reference. To show the grid, right-click in the design area, mouse over **Grid and Guides** and click **Show Grid** as shown:



Note

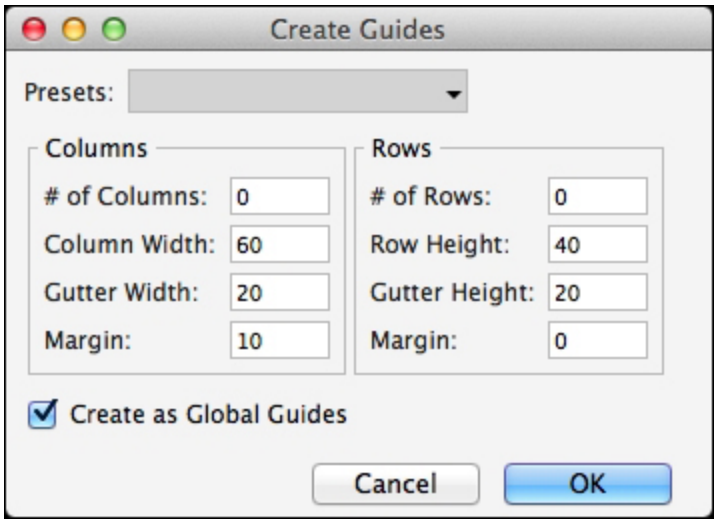
The default spacing for the grid is 10 pixels. To change the default spacing, right-click in the design area, mouse over **Grid and Guides** and click on **Grid Settings…**.

Adding page guides

You can also add page guides to pages in your design. You can select from preset guides or add single page guides. To add page guides, right-click in the design area, mouse over **Grid and Guides** and click on **Create Guides**. In the **Create Guides** dialog window, click on the **Presets** dropdown to select from the following options:

- **960 Grid: 12 Column**
- **960 Grid: 16 Column**
- **1200 Grid: 12 Column**
- **1200 Grid: 15 Column**

You can also create custom guides by changing the default values offered in the **Create Guides** dialog window. The options presented are as follows:



> **Note**
>
> By default, guides are created as global guides. To create a guide just for the current page, uncheck the checkbox next to **Create as Global Guides**.

We can also create single guides for a page. Here are the steps:

1. To create a single guide, click on the left or top ruler, and drag the guide onto the wireframe.
2. To reposition a guide, click and drag the guide to the new position.

> **Note**
>
> By default, single guides are blue and only appear on the current page. To make a global single guide that will be shown on all pages, hold down the *Command* key on a Mac (or the *Ctrl* key on Windows) while clicking and dragging the guide onto the wireframe. By default, global single guides are pink.

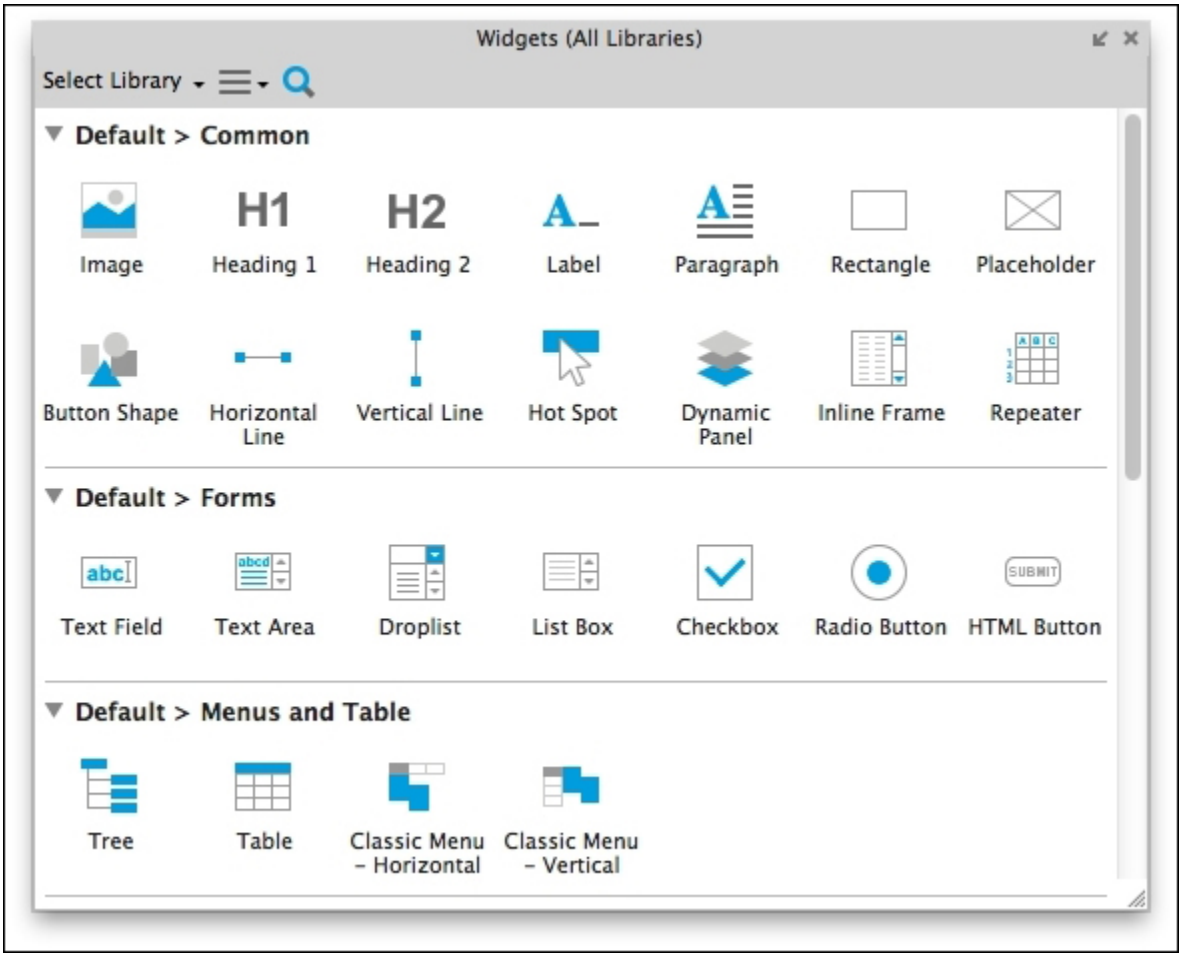
3. Guides can easily be removed as well. To remove a single guide, click on the guide and press *Delete*.
4. To remove all guides, right-click on the design area, mouse over **Grids and Guides**, and select **Delete All Guides**.

Panes

As mentioned briefly, the design area is surrounded to the left, right, and bottom by panes. You will now explore these panes in further detail.

The widgets pane and libraries

In the left-side column under the **Sitemap** pane, you will find the **Widgets** pane as shown here:



The **Widgets** pane allows you to view and use common interface objects also known as widgets (for example, Image, Button Shape, Rectangle, and so on) in your design. Widgets are organized into custom libraries that can be shared and loaded into the **Widgets** pane.

Masters

In the left-hand side column at the bottom, under the **Widgets** pane is the **Masters** pane as shown:

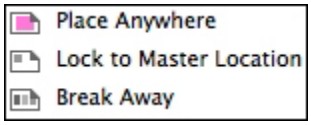


Note

In the **Masters** pane, you will notice **(1)** next to **Masters**. This number indicates the total number of masters available in the opened Axure project file.

The **Masters** pane contains all of the masters available in an RP file. Masters can range from a single widget to a collection of widgets (including interactions) used in your design. In most cases, masters enable you to make changes or updates in a single place. When you update a master, each page that contains the master will be updated as well.

Masters have three types of drop behavior (that is, the master's default behavior for location when placed on the wireframe). Possible drop behavior options are as shown:

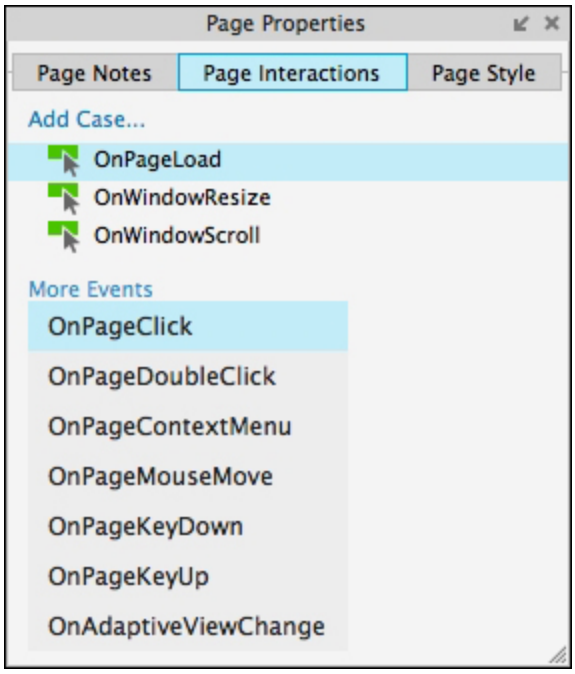


Note

Lock to Master Location means that when a master is placed on a page, the master will always default to the same location as in the original master. **Break Away** masters, when placed on a page, can be edited just like widgets and do not change if the original master is updated.

The Page Properties pane

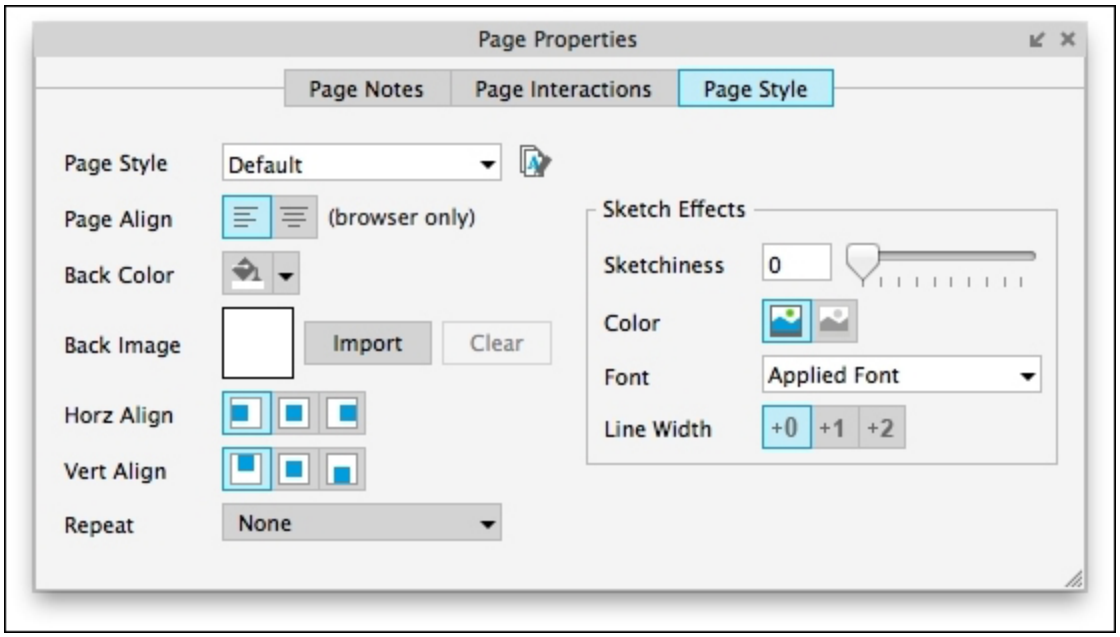
The **Page Properties** pane has three tabs: **Page Notes** (available with Axure RP Pro only), **Page Interactions**, and **Page Style**. The **Page Interactions** tab offers the following interactions as shown:



Note

There are three default events: **OnPageLoad**, **OnWindowResize**, and **OnWindowScroll**. Click on **More Events** to display the additional events shown earlier.

The **Page Style** tab is used to change the style of individual pages as well as page defaults. The **Page Style** tab offers the following options as shown:



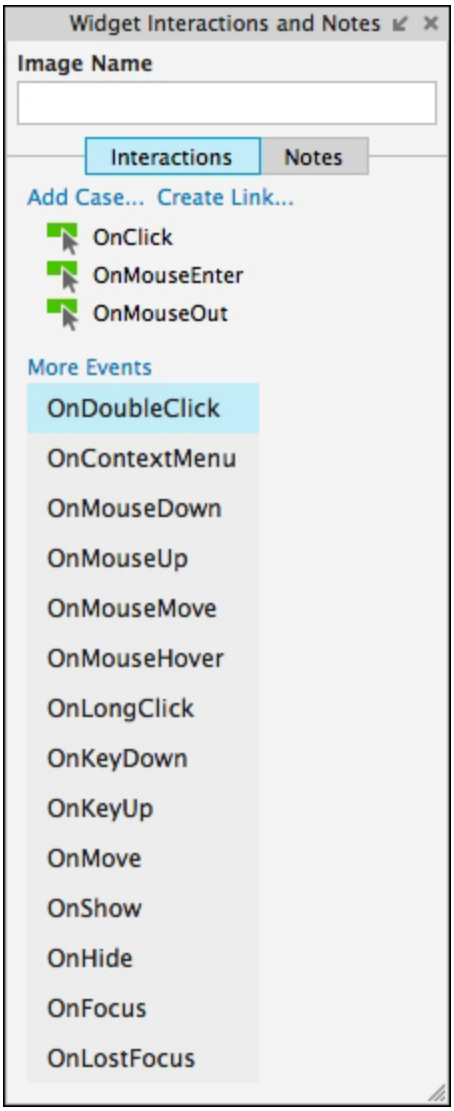
You can use the default page style or define your own custom page styles. The **Page Align** property allows you to left or center align the page (applies to HTML only). You can also set the back color, back image, horizontal alignment, vertical alignment, and whether the back image repeats itself. Sketch effects can also be applied to give the page a handwritten feel.

Note

Learn more about page formatting at https://www.axure.com/learn/basic/page-style.

The Widget Interaction and Notes pane

The **Widget Interaction and Notes** pane has two tabs: **Interactions** and **Notes**. The **Interactions** tab presents interaction options based on the widget selected. An example from an Image widget is shown as follows:



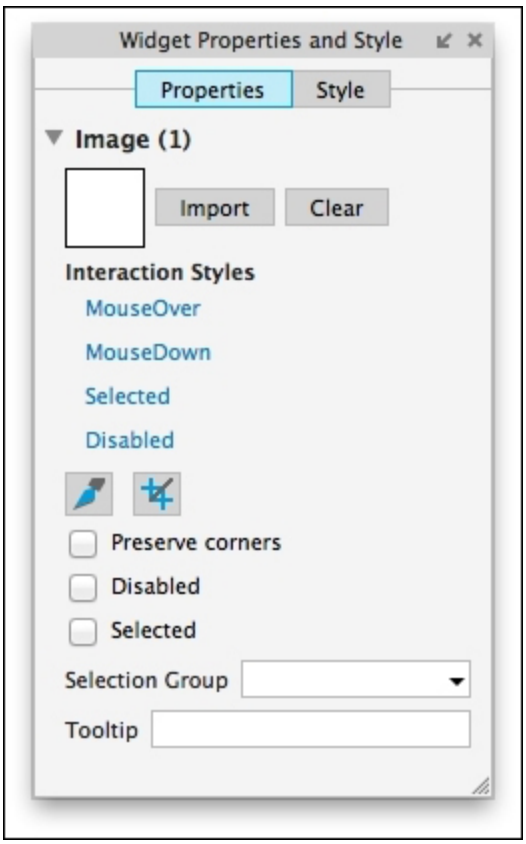
Note

There are three default events: **OnClick**, **OnMouseEnter**, and **OnMouseOut**. Click on **More Events** to display the additional events shown earlier.

The Widget Properties and Style pane

The **Widget Properties and Style** pane has two tabs: **Properties** and **Style**. The **Properties** tab presents interaction options based on

the widget selected. An example from an Image widget is shown as follows:



Interaction styles refer to a widget's visual appearance as a result of an interactive event. For example, we can define different styles when a widget is selected or disabled. We also can define styles for **MouseOver** and **MouseDown** events.

Note

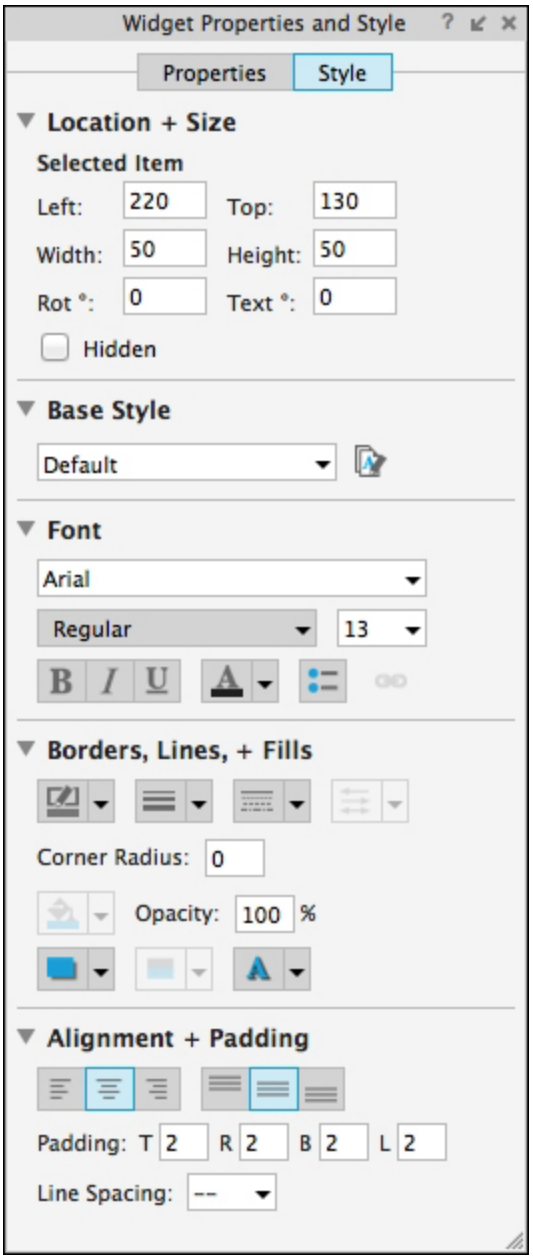
Some widgets will have additional styles and properties. For example, an Image widget's **Widget Properties and Styles** pane has slice (looks like a fountain pen) and crop buttons. The **Widget Properties and Styles** pane for a Paragraph widget would not require this functionality.

Learn more about individual widgets and styles at https://www.axure.com/learn/basic/widgets.

The **Style** tab is used to change the style of the widget selected. The **Style** tab has options organized into the following categories:

- **Location + Size**
- **Base Style**
- **Font**
- **Borders, Lines + Fills**
- **Alignment + Padding**

The **Style** tab is shown as follows:



Note

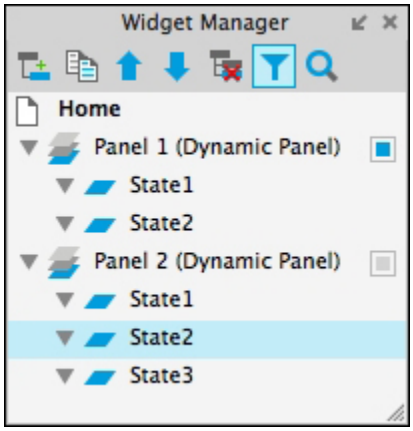
With a widget selected in the design area, we can also access individual widget styles in the Editor toolbar. We can create custom widget styles as well. Under the **Base Style** dropdown, by using the Widget Style Editor (available by clicking on the

icon), you can create custom widget styles as well. Visit https://www.axure.com/learn/basic/widgets to learn more.

The Widget Manager pane

The **Widget Manager** pane is used to change the visibility of dynamic panels as well as manage panel states. Dynamic panels are often used to hide or show content.

The following is an example **Widget Manager** with two dynamic panels labeled **Panel 1** and **Panel 2**:



Note

Next to the dynamic panel labeled **Panel 1**, there is a blue rectangle. This indicates that the default for **Panel 1** is **Show in view**. In contrast, next to the dynamic panel labeled **Panel 2**, there is a gray rectangle. This indicates that the default for **Panel 2** is **Hide from view**. Clicking on the icon will toggle the state.

At the top of the **Widget Manager** pane, you will see a row of seven icons as follows:

- Add State
- Duplicate State

- Move Up
- Move Down
- Delete
- Widget Filter
- Search

Adaptive Views

Axure RP 7 and beyond allow us to create adaptive prototypes with less effort than the previous versions. This is accomplished through a new feature called Adaptive Views. Adaptive Views enable us to create variations of a given screen design. Design variations are optimized for each breakpoint associated with the Adaptive View. Breakpoints are defined based on the browser window's width and/or height in pixels.

Planning for adaptive

If available, we can refer to analytics for our website or application to determine the appropriate width and height to set our breakpoints. When we add an Adaptive View, we are given the opportunity to define new views based on width and/or height, or select from five presets. The preset options are as follows:

- Large display (1200 x any and above)
- Landscape tablet (1024 x by any and below)
- Portrait tablet (768 x any and below)
- Landscape phone (480 x any and below)
- Portrait phone (320 x any and below)

Adaptive Views inherit properties from the parent or base view. The initial design will be created in the base view (or default view). We would then complete additional designs for each child view.

> **Note**
>
> Making changes in child views does not impact the parent views.

There are two primary workflows that individuals and teams leverage when working with Adaptive Views. The workflows and details are as

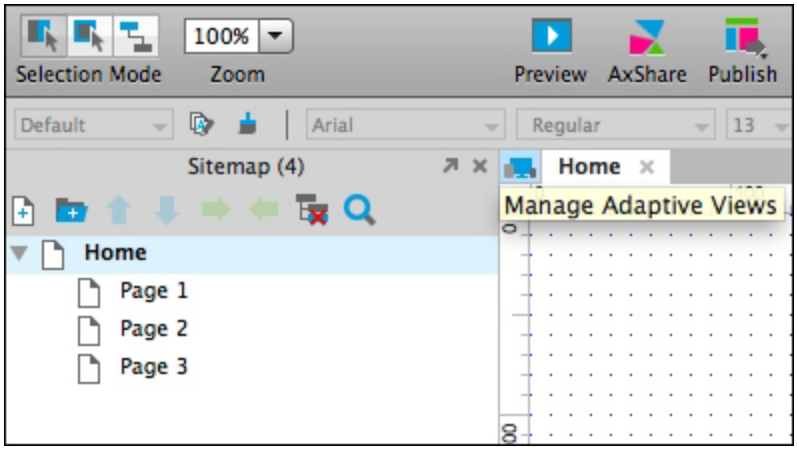
follows:

- Mobile first:
    - The base view is the smallest view
    - Each child view inherits from the previous as the views get larger
- Large to small:
    - The base view is the largest view
    - Each child view inherits from the previous as the views get smaller

Managing Adaptive Views

The following steps will guide you on managing Adaptive Views:

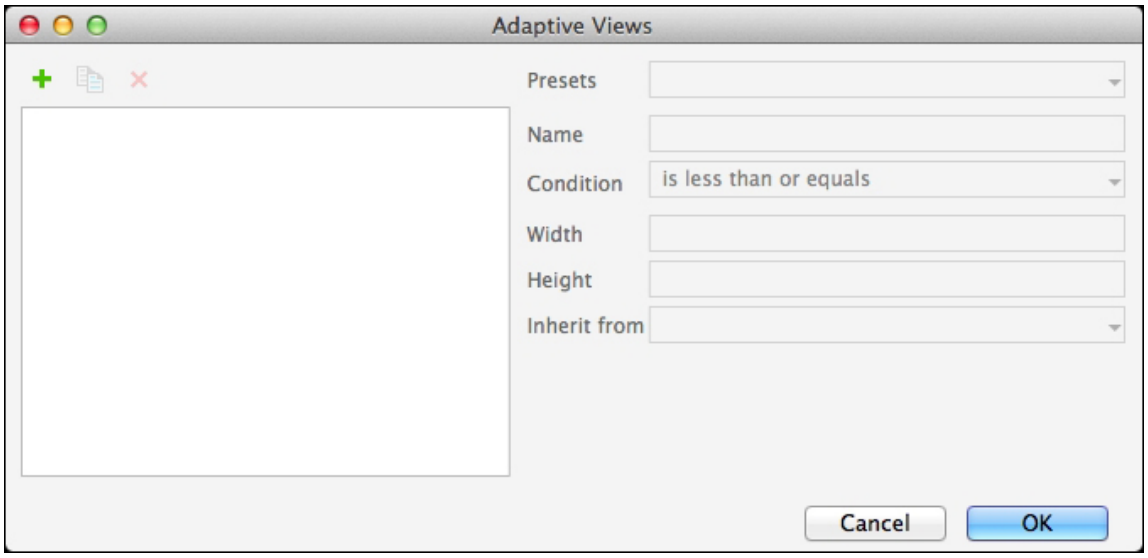
1. To open the **Adaptive Views** dialog box, click on the **Manage Adaptive Views** icon located in the upper-left corner of the design area as shown:



Note

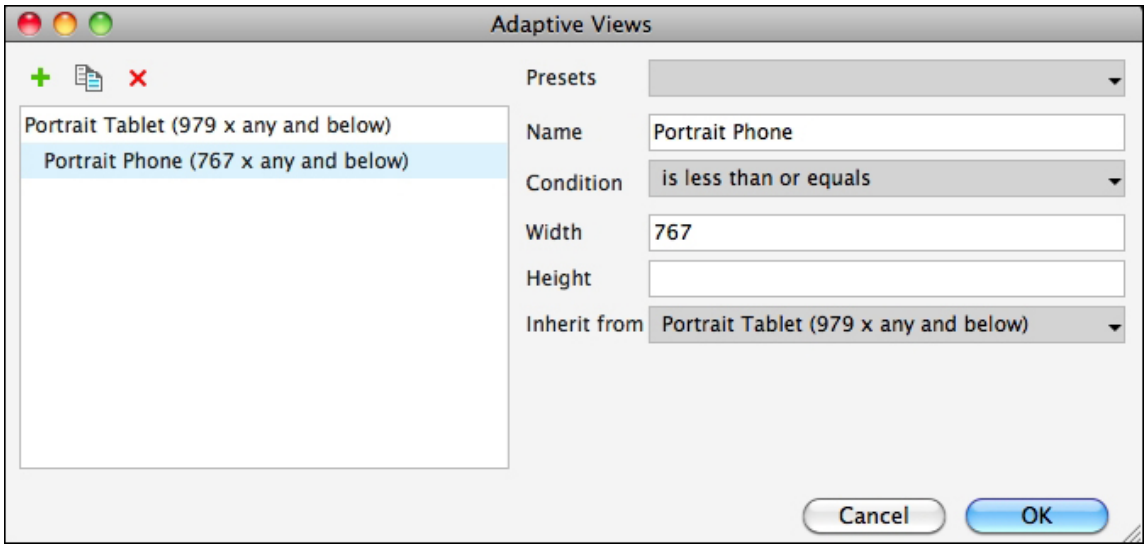
We can also access the **Adaptive Views** dialog box from the main menu (**Project** | **Adaptive Views…**).

The following screenshot shows the default **Adaptive Views** dialog box:

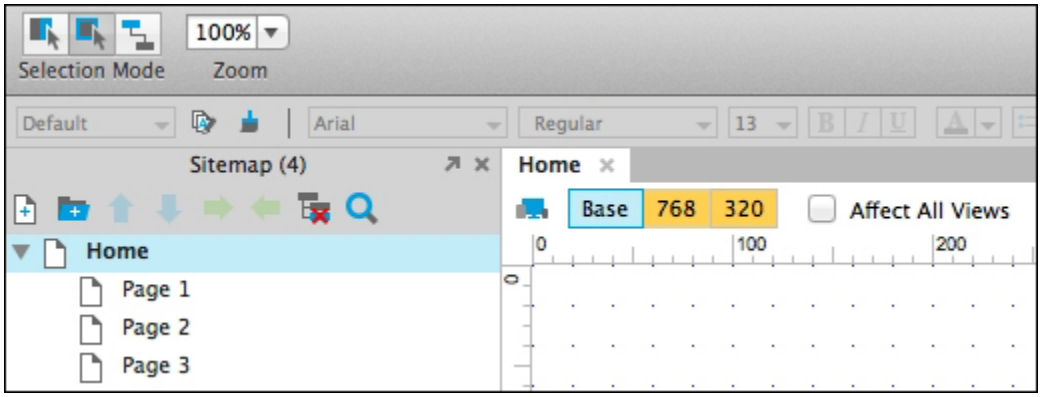


2. Click on the green plus sign to add a new view.
3. In the **Presets** dropdown, select **Portrait Tablet**. Click on the green plus sign to add another new view.
4. In the **Presets** dropdown, select **Portrait Phone**. Once you have added each view, click on the **OK** button to close the **Adaptive View** dialog box.

   The following screenshot shows an **Adaptive Views** dialog box with two views defined:



5. The design area with the portrait tablet and portrait phone views defined would look like this:



We now see three additional tabs in the Adaptive View toolbar corresponding to each of our views: **Base**, **768**, and **320**.

Note

By default, tabs for base or child views are not shown at the top of the design area. Tabs corresponding to Adaptive Views appear once the associated view has been created. Learn more about setting up Adaptive Views at http://www.axure.com/learn/adaptive/setting-up.

Tabs can be in one of four colors: blue, orange, gray, or green. In our example, the **Base** tab is blue, indicating that the view is currently open for editing. The **768** and **320** tabs are orange, indicating that they are child views and could be affected by style changes made on the selected view. If a tab is gray, it indicates that the view will not be affected by style changes (for example, a selected child view would be orange and all the preceding parent views would be gray). Tabs turn green when the **Affect All Views** checkbox is selected, indicating that style changes will impact all views.

Summary

In this chapter, you have learned about the various aspects of Axure's environment and interface. We have explored the main menu, toolbar, sitemap, design area, and panes. We have also become familiar with pages, widgets, and masters. An introduction to Adaptive Views was also explored.

In the next chapter, we will learn how to create common interactions for several widely used web design elements, while building a community site. After organizing these elements into reusable masters, we will design a header, carousel, and a global footer.

Chapter 2. Building a Community Site

Community sites provide individuals who have similar interests with news and information as well as a forum focused on a given subject. Often similar patterns of interactions can be observed when benchmarking community sites. We notice that information is typically grouped into categories containing individual topics presented in a list. A user can click on a category to narrow down the information presented and then click on a given post to view further details.

In this chapter, you will learn about:

- Reviewing the project
    - Background information from our client
- Architecting our community site
    - Inspecting our sitemap
    - Reviewing the forum and news flow
    - Identifying and building our masters
    - Finalizing page design for our community site

Reviewing our project

Under the best circumstances, we are engaged with team members during the initial stages of a project. We may provide input during the scoping of the project, and provide timing and cost estimates that ultimately inform a statement of work. During the discovery phase, we may interview stakeholders to determine the project goals and formulate our approach.

Background information for our client

For our community site, our client is a non-profit startup that will host a *forum for the exploration of Mars*. The client has requested that a prototype be developed showing all the pages of the site. The client would also like to see the use case of a user performing the following tasks:

1. From the **Home** page, clicking on the **Forum & News** page.
2. Clicking on a **Category** header to display a category's topic list.
3. Clicking on a **Topic** and displaying the associated **Topic Detail** page.

The client has also informed us that they want three primary sections on the community site, as follows:

- Our Journey to Mars
- Forum & News
- Universal Charter

The client has also requested a carousel on the home page as well as a social media feed. With these details and additional context provided by the client, we are now ready to begin architecting our community site.

Architecting our community site

We will start by creating a sitemap and generating a flow document. From these items, we can then develop an approach to create the **masters**. Recall that masters provide us with a means to create reusable components. The masters help us to quickly build our pages, while providing a central location to make changes. In general, the changes on a master are automatically reflected in each page that contains the master.

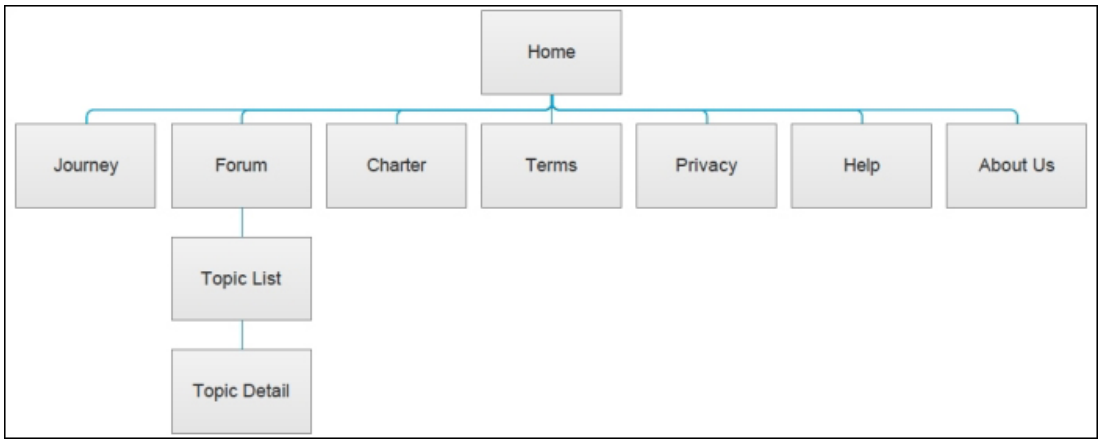
We are now ready to inspect the sitemap and generate the forum and news flow document.

Inspecting our sitemap

The sitemap is used as a tool in site planning and serves as a communication aid. Sitemap shows a site's hierarchy, highlights click paths as well as click depths, and scopes the number of pages needed for a given site.

Based on the background information provided by our client, we built a simple sitemap by adding, naming, and organizing pages in the **Sitemap** pane.

Our sitemap is as follows:



To create your own sitemap in the **Sitemap** pane, perform the following steps:

1. Click on the Add Page icon and type `Sitemap` to name the page.
2. Double-click on the Page icon next to the **Sitemap** page to open the page in the design area.
3. Right-click on the Page icon and click on **Generate Flow Diagram**.
4. In the **Generate Flow Diagram** pop-up screen, select **Standard** and click on **OK**.

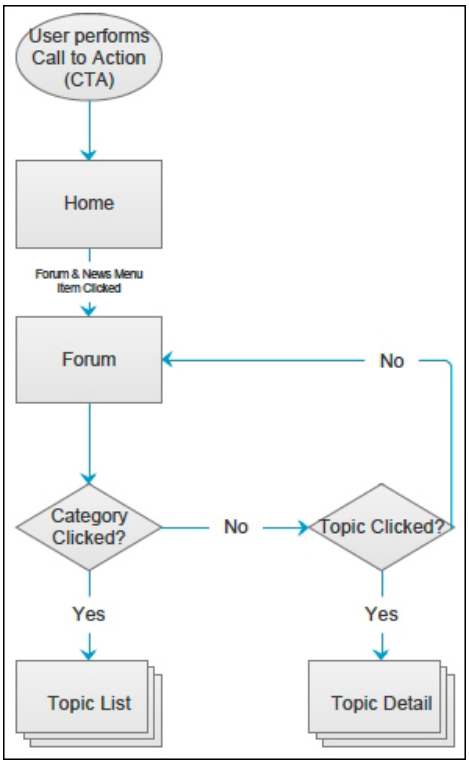
> **Tip**

Downloading the example code

You can download the example code files from your account at http://www.packtpub.com for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit http://www.packtpub.com/support and register to have the files e-mailed directly to you.

Reviewing the forum and news flow

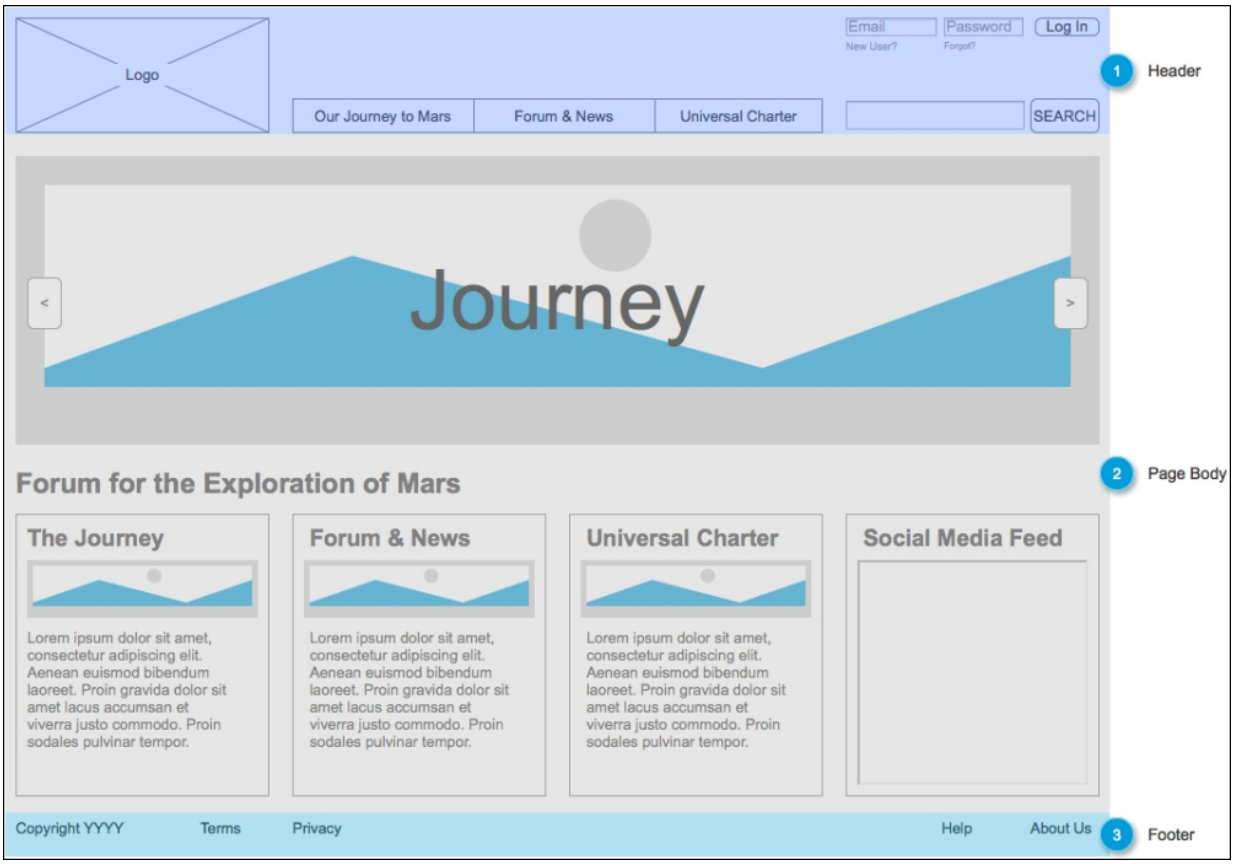
Based on the background information and business requirements provided by our client, we built a flow document using **Flow Widget Library** found in the **Widgets** pane. The flow document depicts the typical pages and decision points for a given use case. Our flow document for the forum and news flow is as follows:



We are now ready to identify and build our masters.

Identifying and building our masters

When identifying widgets as well as interactions best suited for masters, we focus on the basics of web page design. In our design, we will leverage a header, page body, and footer. The following screenshot shows the home page we intend to build with the header, page body, and footer highlighted:



With the page sections identified, we reflect on the logical groupings of individual widgets as well as the commonalities between the interactions. Once we have identified the widgets that will be assembled in a master, we then can create more specified custom interactions by leveraging the raised events. Following this approach, we determine whether we will build the following masters:

- Header
- BreadCrumb
- Secondary_Page_Body
- Informational_Page_Body
- Forum
- Footer

In addition to our masters, we also need to determine which global variables we will need. Based on the project requirements, we will create the following global variables:

- CarouselClicked
- VerticalOffset
- HorizontalOffset
- Index
- BreadCrumbComponent
- Topic
- TopicHeadline

- TopicFilter

> **Note**
>
> We can easily add and remove global variables during future iterations. This, typically, occurs as we refine interactions to represent the user experience of the given page elements, use cases, and flow diagrams.

Next, we will create the global variables as well as the design and interactions for each master.

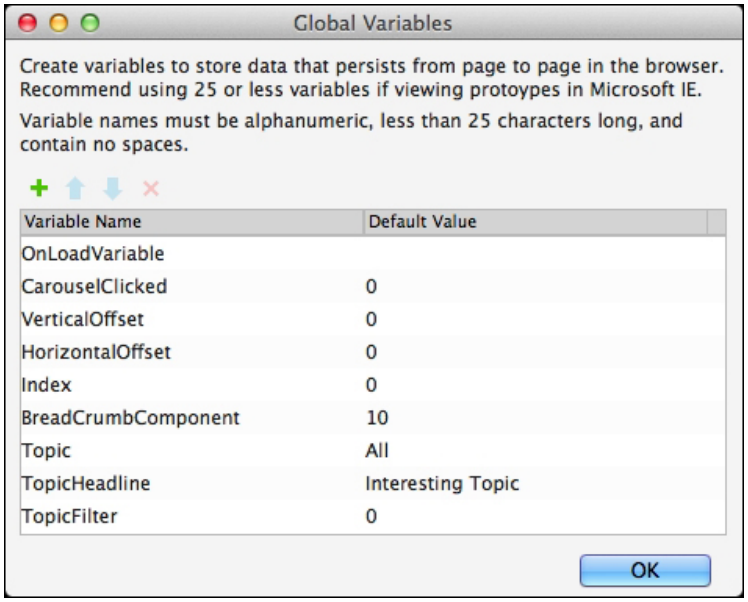
Creating global variables

We will now create the global variables that will allow us to share data between pages in our design. To create the global variables, have a look at the following steps:

1. Navigate to **Project** | **Global Variables…**.
2. In the **Global Variables** dialog, click the green **+** sign and type `CarouselClicked`. Click on the **Default Value** field and type `0`.
3. Repeat Step 2 seven more times to create the additional variables using the following table for the **Variable Name** and **Default Value** fields:

Variable name	Default value
`VerticalOffset`	`0`
`HorizontalOffset`	`0`
`Index`	`0`
`BreadCrumbComponent`	`10`
`Topic`	`All`
`TopicHeadline`	`Interesting Topic`
`TopicFilter`	`0`

4. Once completed, your **Global Variables** dialog will look like this:



5. Click on **OK**.

With our global variables created, we are now ready to complete the design and interactions for each master. We will start by adding masters to the **Masters** pane.

Adding masters to the Masters pane

We will add a total of six masters to the **Masters** pane. To create our masters, perform the following steps:

1. In the **Masters** pane, click on the Add Master icon, type `Header` and press *Enter*.
2. In the **Masters** pane, right-click on the icon next to the **Header** master, then mouse over **Drop Behavior** and click on **Lock to Master Location**.
3. Repeat steps 1-2 five more times to create additional masters using the following table for the master name and drop behavior:

Master name	**Drop behavior**
`BreadCrumb`	Lock to master location
`Secondary_Page_Body`	Lock to master location

Master name	Drop behavior
Informational_Page_Body	Lock to master location
Forum	Place anywhere
Footer	Lock to master location

4. With our masters added to the **Master** pane, we are now ready to complete the design and interactions for each individual master. We will start with the **Header** master.

Designing our Header master

Our **Header** master will be used on all the pages of our community site, as identified in our **Sitemap**. Once completed, the **Header** master will look as follows:



Based on the background information from our client and after the review of the **Widgets** pane, we determine that the **Header** master will be made up of the following widgets:

Quantity	Widget type
1	Placeholder
3	Text Field
2	Button Shape
2	Label
1	Classic Menu - Horizontal

With the quantity and type of widgets identified, we are now ready to place the widgets on the **Header** master.

Placing widgets on the Header master

All of the widgets used in the **Header** master can be found in the **Default** | **Common, Default** | **Forms and Default** | **Menus and Table** dropdown of the widgets library. To place the widgets on the **Header** master, perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Header** master.
2. In the **Widgets** pane, drag the **Text Field** widget and place at the coordinates (730, 10).
3. Select the **Text Field** widget.
4. Type `Email`. In the toolbar, change **w:** to `79` and **h:** to `16`.
5. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `EmailTextField`.
6. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Font** and perform the following steps:
    1. Change the font size to `11`.
    2. Click the down arrow next to the Text Color icon. In the drop-down menu, go in the **#** text field and enter `999999`.
7. Repeat steps 2-6 to complete the design of the **Header** master using the values listed in the following table (* if applicable):

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)	Font size*	Color* (In Hex #)
Placeholder	(10,10)	`Logo`	`220`	`100`	`CommunityLogo`		
Text Field	(815,10)	`Password`	`69`	`16`	`PasswordTextField`	`11`	`999999`
Button Shape	(894,10)	`Log In`	`56`	`16`	`LogInButton`		

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)	Font size*	Color* (In Hex #)
Label	(730,30)	New User?	45	10	NewUserLink	9	999999
Label	(815,30)	Forgot?	27	9	ForgotLink	9	999999
Text Field	(730,82)		155	25	SearchTextField		
Button Shape	(890,80)	Search	60	30	SearchButton		
Classic Menu - Horizontal	(250,80)		460	30	HzMenu		

To complete the design of the **Header** master, we need to name the menu items on **HzMenu**. With **HzMenu** selected, perform the following steps:

1. Click on the first menu item and type `Our Journey to Mars`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `JourneyMenuItem`.
2. Click on the first menu item and type `Forum & News`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `ForumMenuItem`.
3. Click on the first menu item and type `Universal Charter`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `CharterMenuItem`.
4. With the design completed for the **Header** master, we are now ready to define the interactions.

Building interactions for our Header master

Interactions with our **Header** master will be triggered by the following named widgets and events:

Named widget	Event(s)
CommunityLogo	**OnClick**
HzMenu	**OnClick**
EmailTextField	**OnMouseEnter** **OnMouseOut**
PasswordTextField	**OnMouseEnter** **OnMouseOut**

We will now define the interactions for each widget starting with the `CommunityLogo`.

Defining interaction for CommunityLogo

To define the **OnClick** interaction, please have a look at the following steps:

1. Click on the **CommunityLogo** name field at the coordinates (10,10).
2. In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Create Link…**.
3. In the pop-up sitemap, click on **Home**.

Next, we will enable the interactions for **HzMenu**.

Enabling interactions for HzMenu

To enable the **OnClick** interactions for each menu item of **HzMenu**, click on the first menu item (that is, **JourneyMenuItem**) at the coordinates (250,80) and perform the following steps:

1. To enable interaction for **JourneyMenuItem**, in the **Widget Interactions and Notes** pane with the **Interactions** tab selected click on **Create Link…**. In the pop-up sitemap, click on **Journey**.
2. Click on the second menu item (that is, **ForumMenuItem**). To enable interaction for **ForumMenuItem,** in the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Create Link…**. In the pop-up sitemap, click on **Forum**.
3. Click on the third menu item (that is, **CharterMenuItem**). To enable interaction for **CharterMenuItem**, in the **Widget Interactions and Notes** pane with the

Interactions tab selected, click on **Create Link…**. In the pop-up sitemap, click on **Charter**.

Next, we will enable the interactions for **EmailTextField**.

Illuminating interactions for EmailTextField

Once a text field has been clicked, a user expects focus to remain on the text field until the user moves the cursor off the text field. Under normal conditions and in most prototyping scenarios, this would not be an issue that we would have to consider.

Since our community site leverages a carousel, we will implement a method that ensures that the carousel stops autorotation when the user clicks on the **Previous** or **Next** buttons. This method leverages a **Set Focus** event on a hidden Hot Spot named **CheckForClick**. The **Set Focus** event on **CheckForClick** also occurs when the carousel dynamic panel changes state. If the **Previous** or **Next** buttons are clicked, the **CarouselClicked** global variable will be set to `1`. The **CheckForClick OnFocus** event's **Case 1** will only execute if **CarouselClicked** is `0`. This is how we keep the carousel from advancing, if the user clicks on the **Previous** or **Next** button during a wait state.

In order to keep focus on a text field under these conditions, we leverage the **OnMouseEnter** event and set the value of **CarouselClicked** to **FieldActive** only if **CarouselClicked** does not have a value of 1. This ensures that the focus will not be set on **CheckForClick** as long as the user keeps the mouse over the text field.

Once the user removes the mouse from the text field, the **OnMouseOut** event is triggered. If **CarouselClicked** does not have a value of 1, we set the value of **CarouselClicked** to `0` and trigger a raised event specific to that given text field. This enables us to create unique interactions for each raised event on a per page basis.

We will now define the **OnMouseEnter** and **OnMouseOut** events for **EmailTextField**.

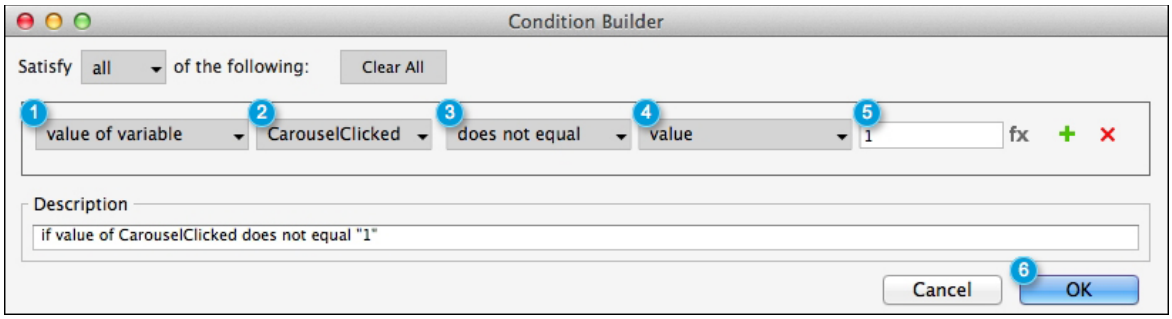
Activating the OnMouseEnter event

To activate the **OnMouseEnter** event, perform the following steps:

1. Click on **EmailTextField** at the coordinates (730,10).
2. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, then click on **More Events**, and double-click **OnMouseEnter**. A **Case Editor** dialog box will open.
3. In the **Case Editor** dialog, perform these steps:

Add the condition as follows:

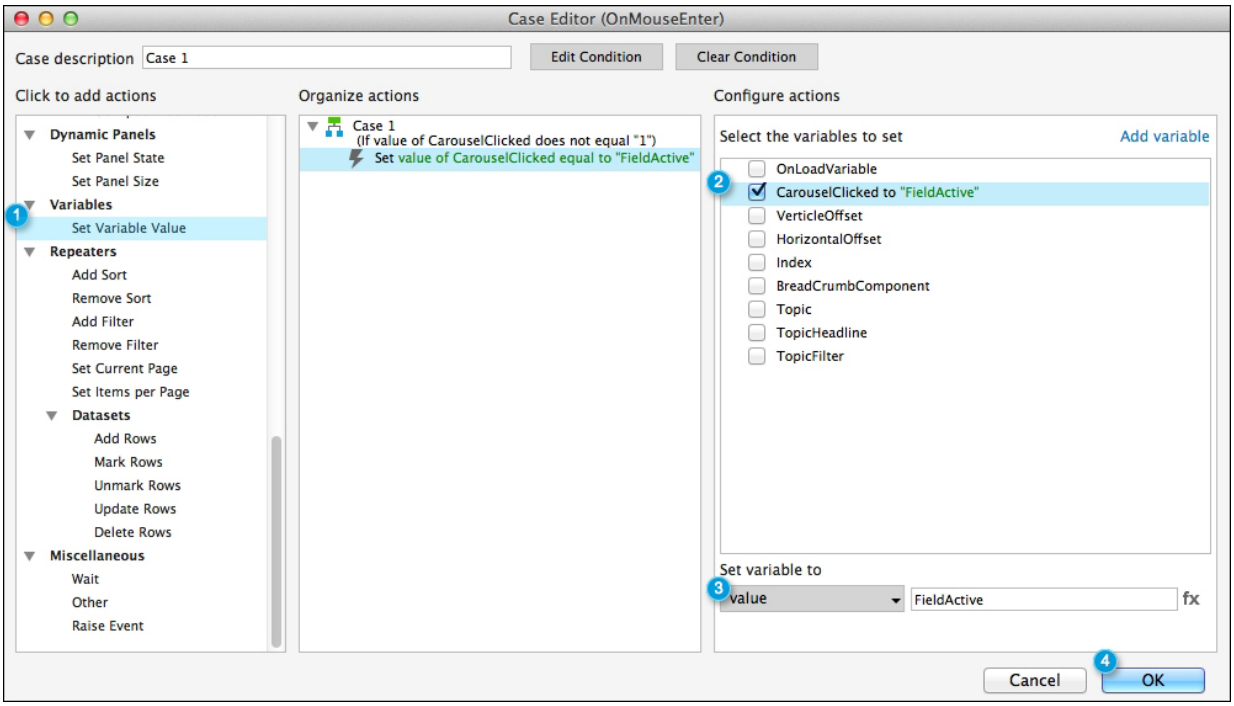
1. Click on the **Add Condition** button.
2. You will see the **Condition Builder** dialog box as shown here:



As shown in the previous screenshot, perform the following steps in the outlined condition box:

1. In the first dropdown, select **value of variable**.
2. In the second dropdown, select **CarouselClicked**.
3. In the third dropdown, select **does not equal**.
4. In the fourth dropdown, select **value**.
5. In the text field, enter `1`.
6. Click on **OK**.

To create the action, set the value of the **CarouselClicked** variable using these steps:



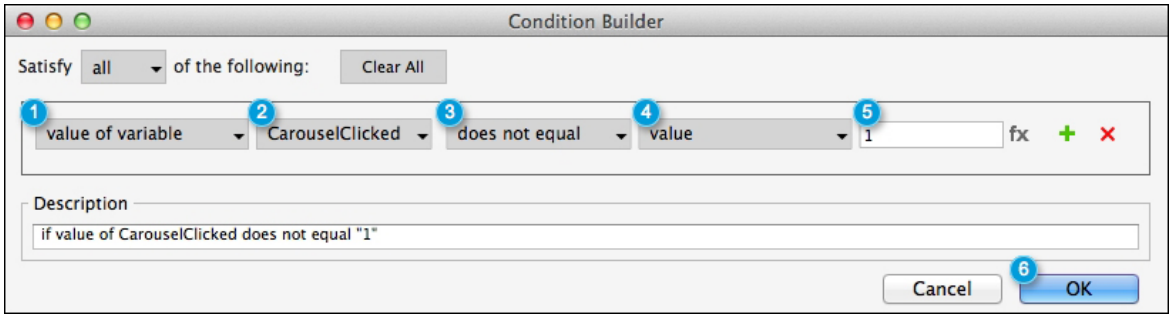
1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
2. Under **Configure actions**, click on the checkbox next to **CarouselClicked**.
3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `FieldActive`.
4. Click on **OK**.

Next, we will define the **OnMouseOut** event for **EmailTextField**.

Facilitating the OnMouseOut event

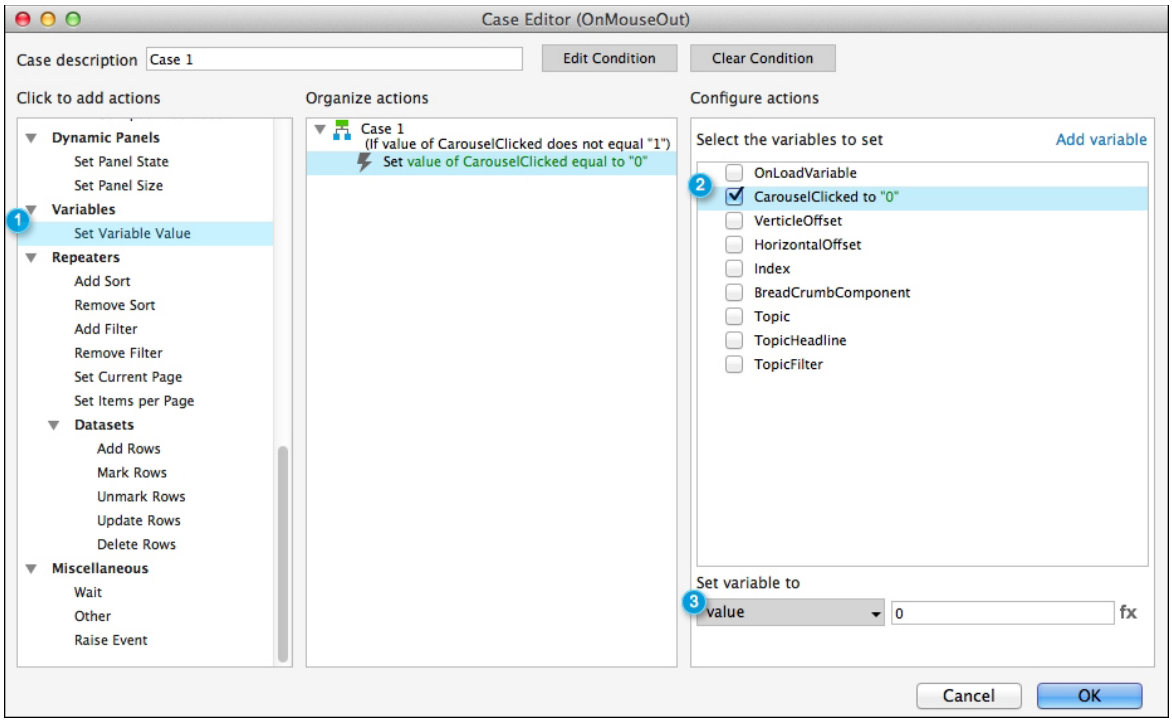
To activate the **OnMouseOut** event, perform the following steps:

1. Click on **EmailTextField** at the coordinates (730,10).
2. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, then click on **More Events**, and double-click on **OnMouseOut**.
3. A **Case Editor** dialog box will open. In the **Case Editor** dialog, click on the **Add Condition** button to add the condition.
4. The **Condition Builder** dialog box will pop up. Perform the steps in the outlined condition box, as shown in the following screenshot:



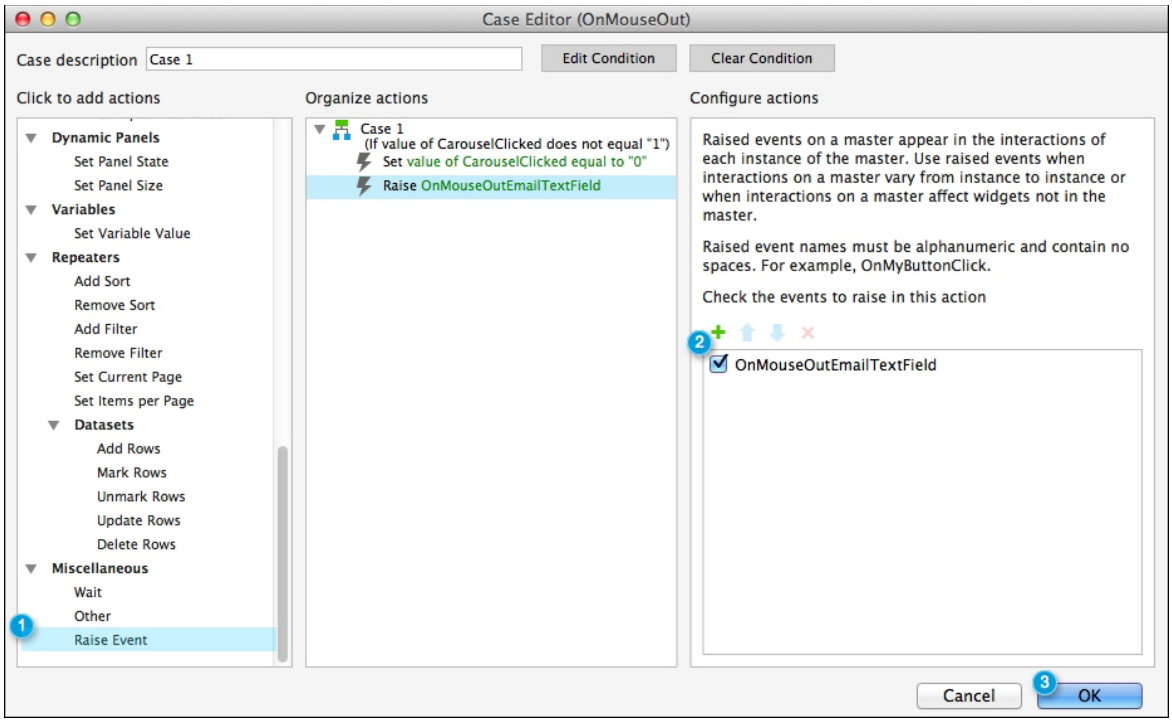
   1. In the first dropdown, select **value of variable**.
   2. In the second dropdown, select **CarouselClicked**.
   3. In the third dropdown, select **does not equal**.
   4. In the fourth dropdown, select **value**.
   5. In the text field, enter `1`.
   6. Click on **OK**.

5. Now to create the first action, set the value of the **CarouselClicked** variable and perform the following steps:



1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
2. Under **Configure actions**, click on the checkbox next to **CarouselClicked**.
3. Under **Set variable to**, click the first dropdown and click on **value**. In the text field, enter `0`.

6. To create the second action, that is, the **OnMouseOutEmailTextField** raised event, perform the following steps:



1. Under **Click to add actions**, scroll to the **Miscellaneous** drop-down menu and click on **Raise Event**.
2. Under **Configure actions**, click on the green **+** and type **OnMouseOutEmailTextField**. Click on the checkbox next to **OnMouseOutEmailTextField**.
3. Click on **OK**.

With the interactions defined for **EmailTextField**, we are now ready to define interactions for **PasswordTextField**.

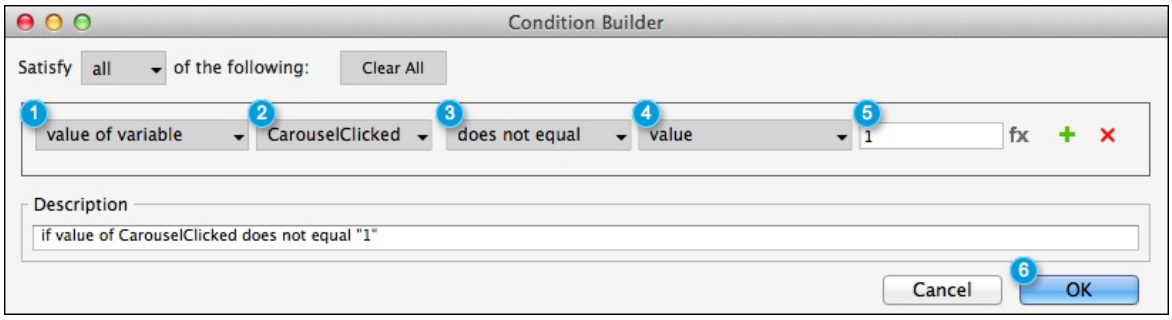
Constructing interactions for the PasswordTextField

We will now construct the interactions for the **OnMouseEnter** and **OnMouseOut** events in **PasswordTextField**.

Activating the OnMouseEnter event

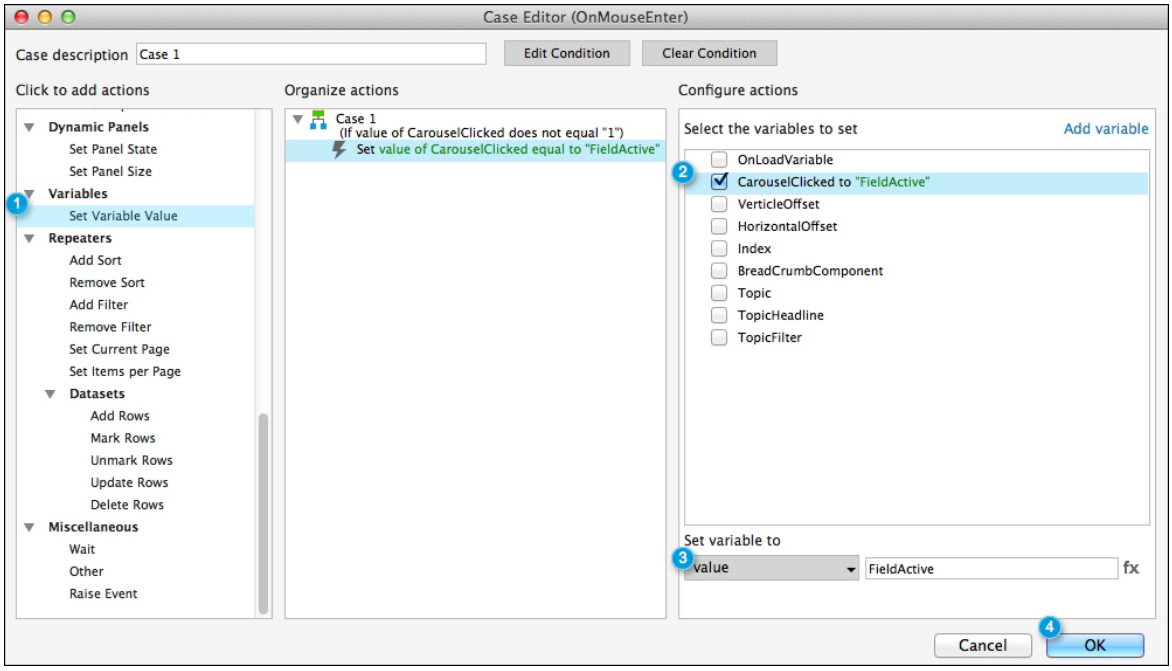
To activate the **OnMouseEnter** event, perform the following steps:

1. Click on **PasswordTextField** at the coordinates (815,10).
2. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, click on **More Events**, and then double-click on **OnMouseEnter**.
3. A **Case Editor** dialog box will open. In the **Case Editor** dialog, click on the **Add Condition** button.
4. The **Condition Builder** dialog box will open. Perform the following steps in the outlined condition box, as shown in the following screenshot:



1. In the first dropdown, select **value of variable**.
2. In the second dropdown, select **CarouselClicked**.
3. In the third dropdown, select **does not equal**.
4. In the fourth dropdown, select **value**.
5. In the text field, enter `1`.
6. Click on **OK**.

5. Now to create the action, set the value of the **CarouselClicked** variable by performing the following steps:



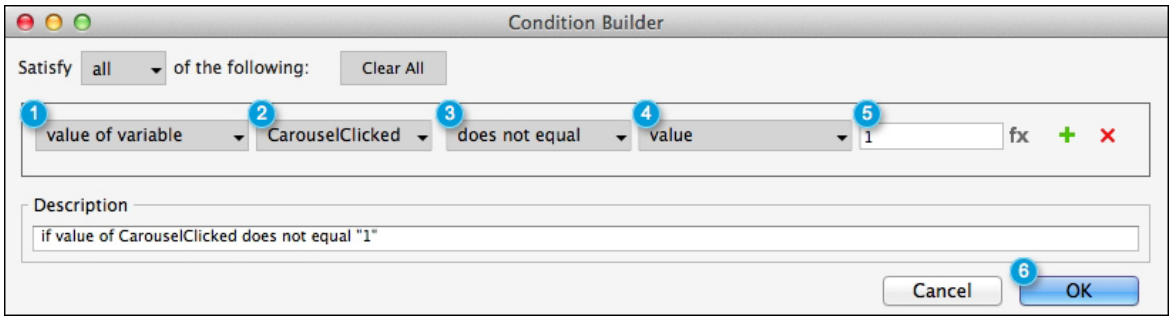
1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
2. Under **Configure actions**, click on the checkbox next to **CarouselClicked**.
3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `FieldActive`.
4. Click on **OK**.

Next, we will define the **OnMouseOut** event for **PasswordTextField**.

Facilitating the OnMouseOut Event

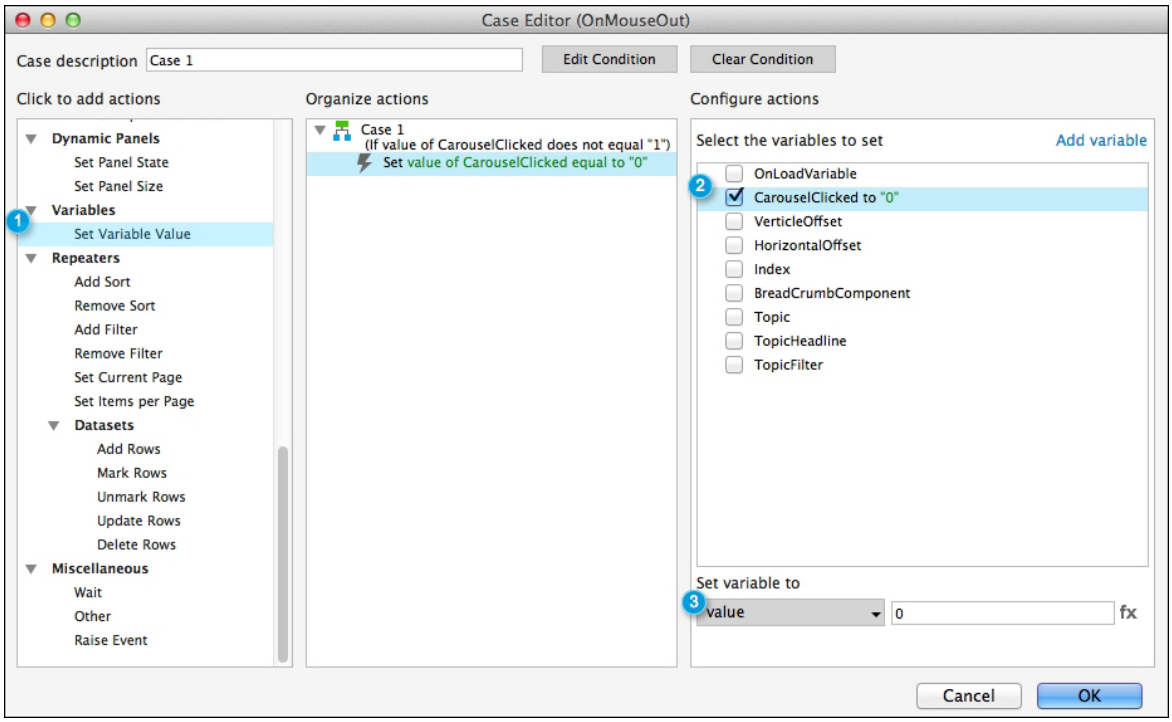
To activate the **OnMouseOut** event, perform the following steps:

1. Click on **PasswordTextField** at the coordinates (815,10).
2. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, then click on **More Events**, and double-click on **OnMouseOut**.
3. A **Case Editor** dialog box will open. In the **Case Editor** dialog, click on the **Add Condition** button to add the condition.
4. In the **Condition Builder** dialog box as shown in the following screenshot, perform the following steps in the outlined condition box:



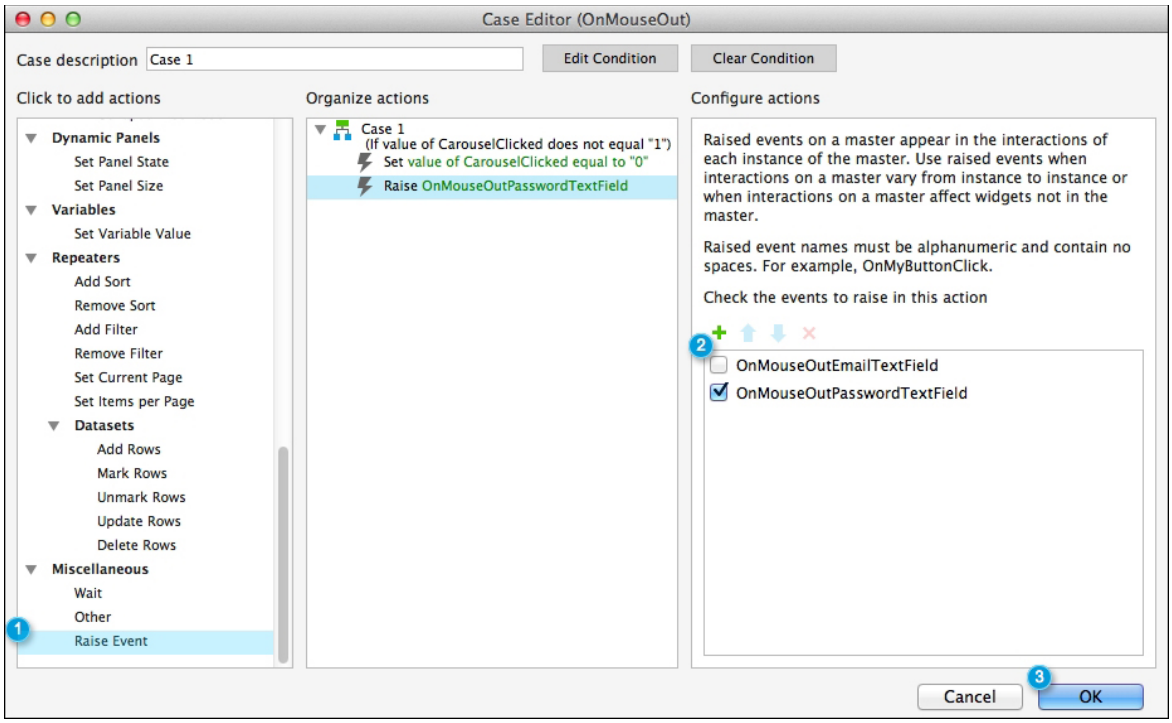
   1. In the first dropdown, select **value of variable**.
   2. In the second dropdown, select **CarouselClicked**.
   3. In the third dropdown, select **does not equal**.
   4. In the fourth dropdown, select **value**.
   5. In the text field, enter `1`.
   6. Click on **OK**.

5. Now to create the first action, set the value of the **CarouselClicked** variable by performing the following steps:



1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
2. Under **Configure actions**, click on the checkbox next to **CarouselClicked**.
3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `0`.

6. To create the second action, that is, the **OnMouseOutPasswordTextField** raised event, perform the following steps:



1. Under **Click to add actions**, scroll to the **Miscellaneous** drop-down menu and click on **Raise Event**.
2. Under **Configure actions**, click on the green **+** and type **OnMouseOutPasswordTextField**. Click on the checkbox next to **OnMouseOutPasswordTextField**.
3. Click on **OK**.

With the design and interactions completed for the **Header** master, we are now ready to construct our **BreadCrumb** master.

Constructing our BreadCrumb master

Wayfinding helps users navigate virtual spaces and provides a proper orientation when navigating websites. A common design paradigm that provides navigational context is the use of a **BreadCrumb** master.

We will construct a dynamic **BreadCrumb** master that will be used on every page of our community site. Once completed, the **BreadCrumb** master will look as follows:

```
Home  >  Forum & News  >  Interesting Topic
```

We will construct a dynamic **BreadCrumb** master using the following widgets:

Quantity	Widget type
1	Repeater
1	Dynamic Panel
1	Label

We will now format a Repeater and place the widgets on the **BreadCrumb** master.

Formatting the Repeater widget and placing widgets on the BreadCrumb master

All of the widgets used in the **BreadCrumb** master can be found in the **Default |Common** dropdown of the widgets library. To format the **BreadCrumb** Repeater and place the widgets in the **BreadCrumb** master, perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **BreadCrumb** master to open in the design area.
2. In the **Widgets** pane, click on the **Repeater** widget. Drag the Repeater widget onto the design area and place at the coordinates (10,113).
3. With the **Repeater** widget selected, in the **Widget Interactions and Notes** pane click in the **Repeater Name** field and type `BreadCrumb`.
4. Double-click on the **BreadCrumb** Repeater widget to open it in the design area.
5. Under the design area in the Repeater pane, click the **Repeater Style** tab. To change the default Repeater style, perform the following steps:
    1. Under the **Layout** dropdown, click on the radio button next to **Horizontal**.
    2. Under the **Spacing** dropdown, in the **Spacing: Row** field enter `5`.
6. Under the design area in the Repeater pane, click on the **Repeater Dataset** tab. To change and add a column to the **BreadCrumb** Repeater, perform the following steps:
    1. Double-click on **Column0** and type `BreadCrumbComponent`.
    2. Double-click on **Add Column** and type `PageLink`.
7. Click on the **Rectangle** widget at the coordinates (0,0) and delete it.
8. From the **Widgets** pane, drag a **Label** widget on to the design area and place at the coordinates (0,8).
9. With the **Label** widget selected, perform the following steps:
    1. Type `A`. In the toolbar, change **w:** to `7` and **h:** to `15`.

2. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `LinkLabel`.
3. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Font**. In the first dropdown, change the font family to **Courier New** and change the font size to `10`.

10. Right-click on the **Label** widget and click on **Convert to Dynamic Panel**. With the new dynamic panel widget selected, perform the following steps:

    1. In the toolbar, change **w:** to `10` and **h:** to `15`.
    2. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `LinkDP`.
    3. In the **Widget Manager** pane, slow-double-click the **State1** state name and rename it to `Link`.
    4. In the **Widget Manage**r pane, click on the state icon beside **Link** two times to open the Link state in the design area.

11. With the **Link** state opened in the design area, click on the **LinkLabel** widget at the coordinates (0,0). In the toolbar, change **x:** to `3`.

With the design completed for the **BreadCrumb** master, we are now ready to update the Repeater item on the **BreadCrumb** repeater.

Updating the Repeater item's OnItemLoad event for the BreadCrumb repeater

The Repeater item for our **BreadCrumb** repeater contains the **LinkDP** dynamic panel with the **Link** state. Recall the **Link** state has only one Label widget called **LinkLabel**. When the **OnItemLoad** event of the **BreadCrumb** repeater is triggered, the following actions will be performed:

- Set text on **LinkLabel**
- Set variable value of the **BreadCrumbComponent** global variable
- Set panel size of the **LinkDP** dynamic panel
- Move the **LinkDP** dynamic panel
- Set variable value of the **HorizontalOffset** global variable

As the repeater iterates, we update the text on the **LinkLabel** to display the current link in the **BreadCrumb**. Next, we set the **BreadCrumbComponent** global variable to the current **Item.BreadCrumbComponent** value in our repeater. We then use the **BreadCrumbComponent** to calculate the string length to assist in setting the size of the **LinkDP** dynamic panel.

In addition, we move the **LinkDP** dynamic panel based on **HorizontalOffset**. Finally, we update the **HorizontalOffset** global variable based on the length of the **BreadCrumbComponent** multiplied by a pixel constant.

Note

Most fonts are variable-width. We selected the fixed-width (that is, monospaced) font, Courier New, to minimize the inconsistent spacing variation between the links and

delimiters in our dynamic **BreadCrumb**.

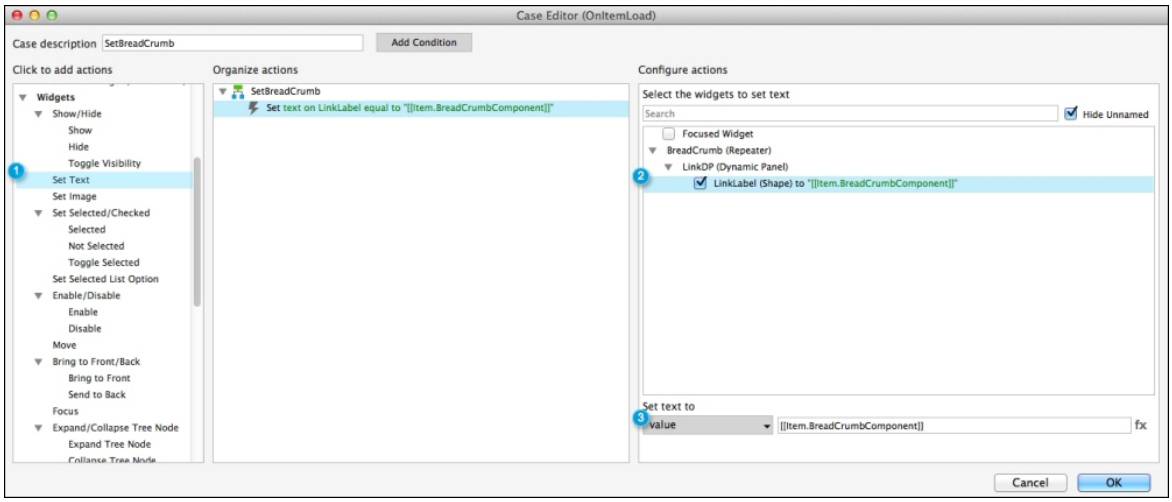
We will now define the **OnItemLoad** event actions for the **BreadCrumb** repeater.

Defining the OnItemLoad event actions

Reopen the **BreadCrumb** master in the design area and click on the **BreadCrumb** repeater at the coordinates (10,113). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, click on **More Events**, and double-click on **OnItemLoad**. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

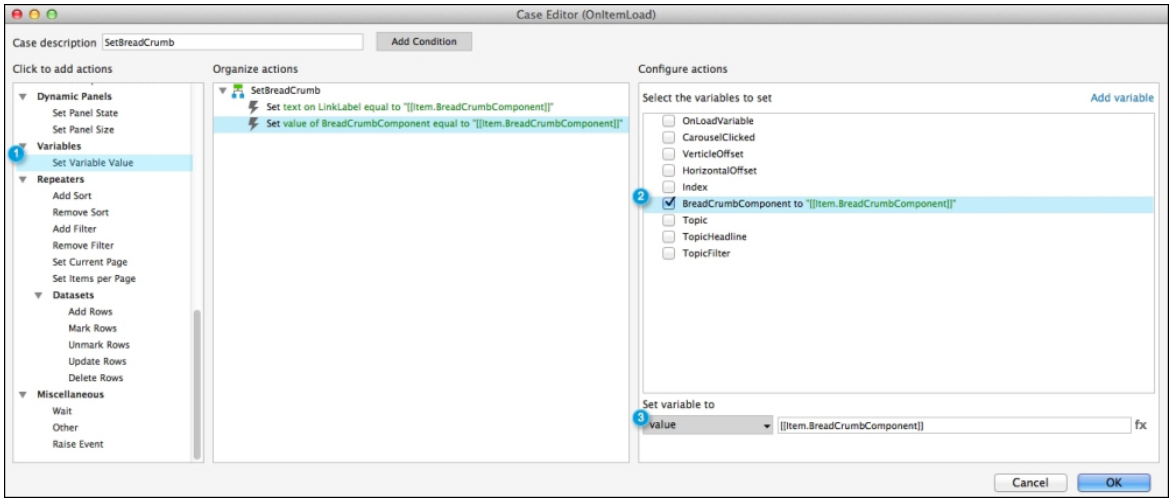
1. In the **Case description** field, type `SetBreadCrumb`.
2. Create the first action to set the text on **LinkLabel** by performing the following steps:

.



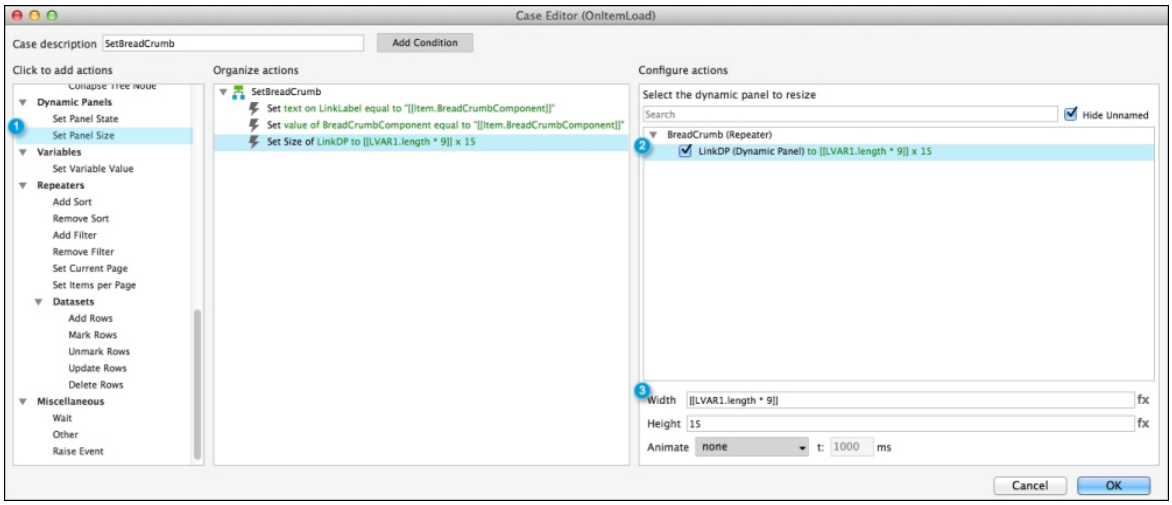
   1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
   2. Under **Configure actions**, click the checkbox next to **LinkLabel**.
   3. Under **Set text to**, click the first dropdown and click **value**. In the text field, enter `[[Item.BreadCrumbComponent]]`.

3. Create the second action to set the value of the **BreadCrumbComponent** variable using the following steps:

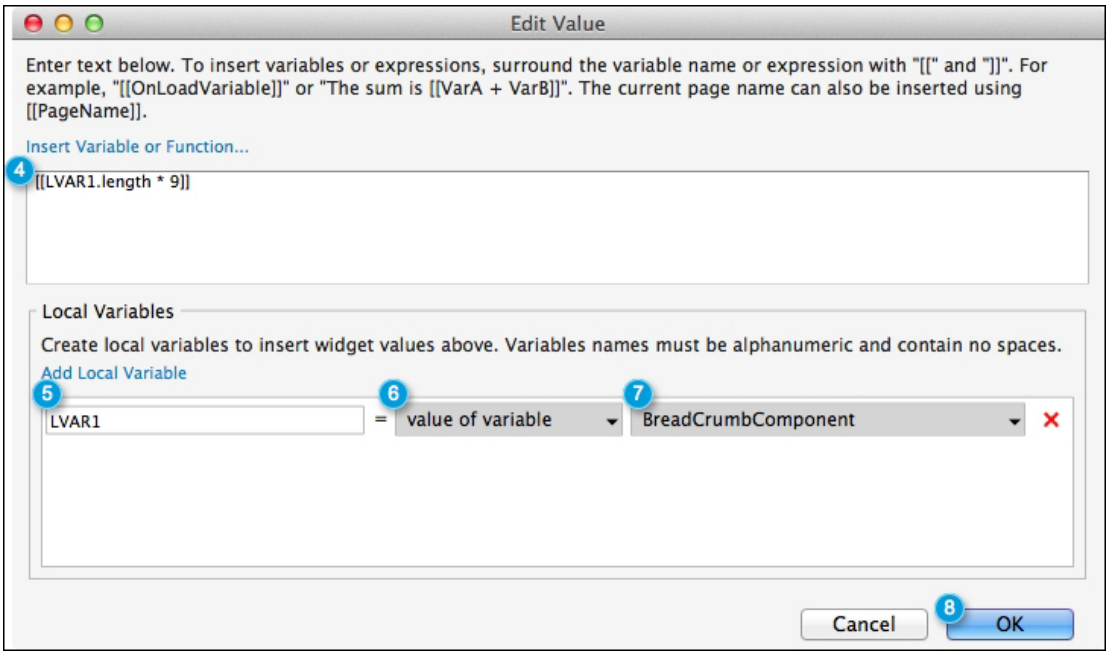


1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
2. Under **Configure actions**, click the checkbox next to **BreadCrumbComponent**.
3. Under **Set variable to**, click the first dropdown and click **value**. In the text field, enter `[[Item.BreadCrumbComponent]]`.

4. Create the third action to set the size of the **LinkDP** dynamic panel by performing the following steps:

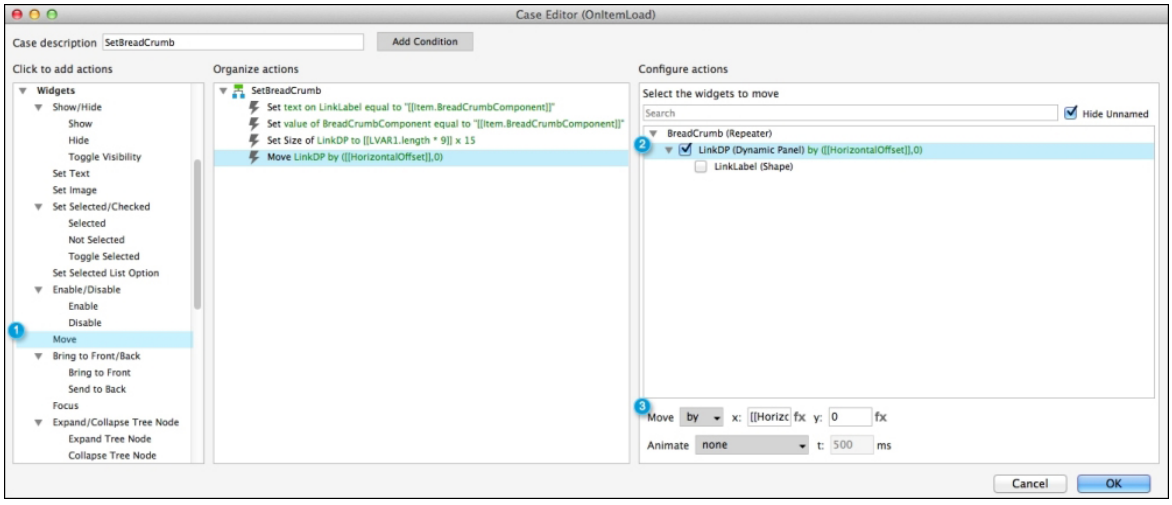


1. Under **Click to add actions**, scroll to the **Dynamic Panels** drop-down menu and click on **Set Panel Size**.
2. Under **Configure actions**, click the checkbox next to **LinkDP**.
3. Beside the **Width** text field, click the fx icon to open the **Edit Value** dialog box. In the **Edit Value** dialog box, perform steps 4-8. Have a look at the following screenshot to understand better:



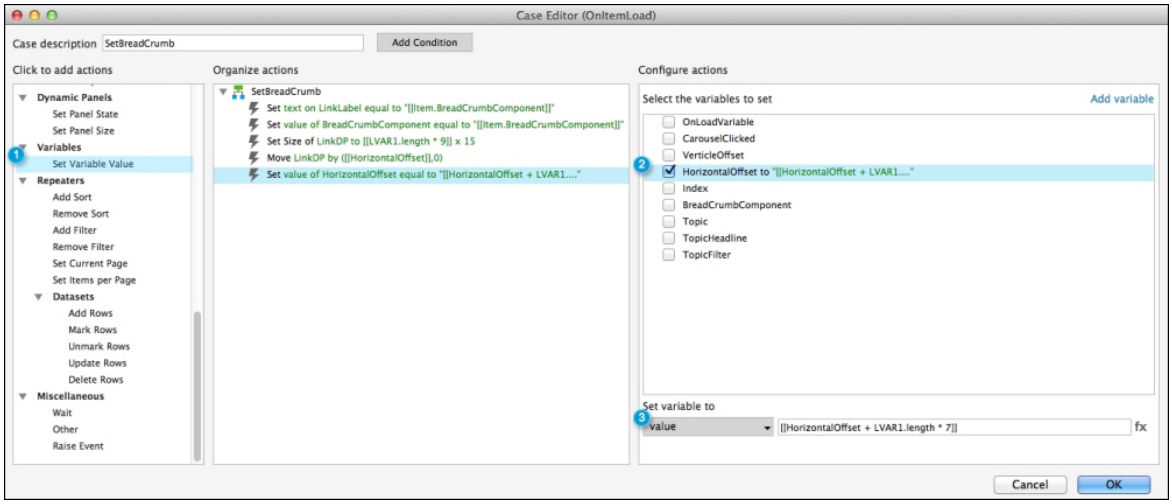
4. Under **Insert Variable or Function…**, type `[LVAR1.length * 9]]`.
5. Beneath **Local Variables** under **Add Local Variable**, in the text field type `LVAR1`.
6. In the first dropdown, select **value of variable**.
7. In the second dropdown, select **BreadCrumbComponent**.
8. Click on **OK**.

5. Create the fourth action to move the **LinkDP** dynamic panel using the following steps:

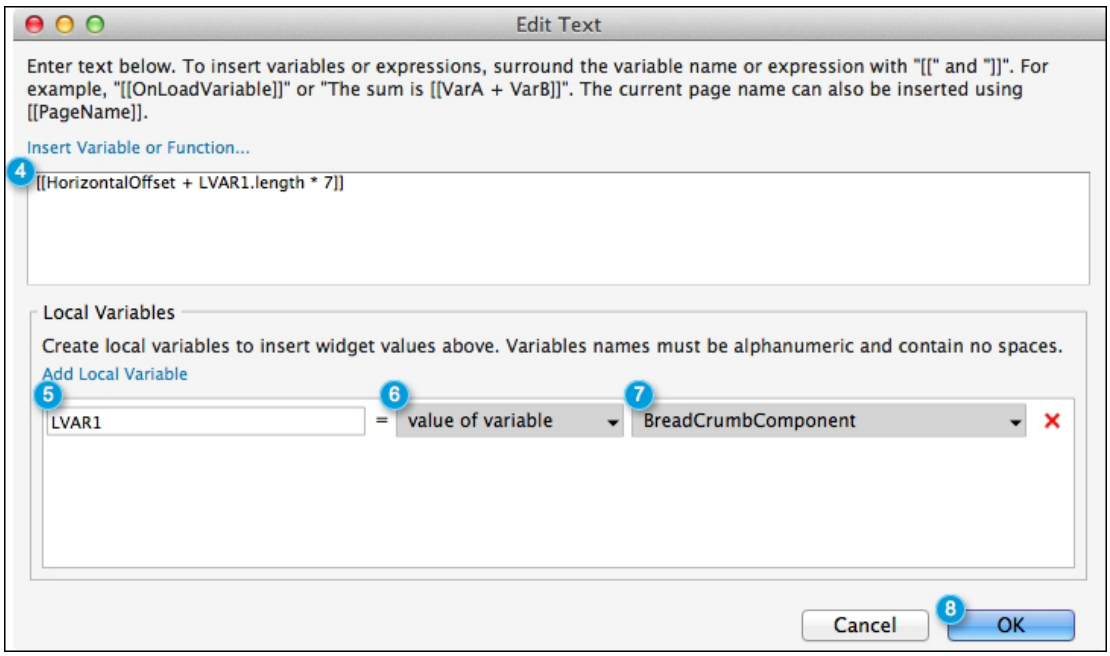


1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Move**.
2. Under **Configure actions**, click the checkbox next to **LinkDP**.
3. Next to **Move**, click the first dropdown and click on **by**. In the text field, enter `[[HorizontalOffset]]`.

6. Create the fifth action to set value of the **HorizontalOffset** variable by performing the following steps:



1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
2. Under **Configure actions**, click on the checkbox next to **HorizontalOffset**.
3. Beside the **Width** text field, click on the fx icon to open the **Edit Value** dialog box. In the **Edit Value** dialog box, perform steps 4-8.



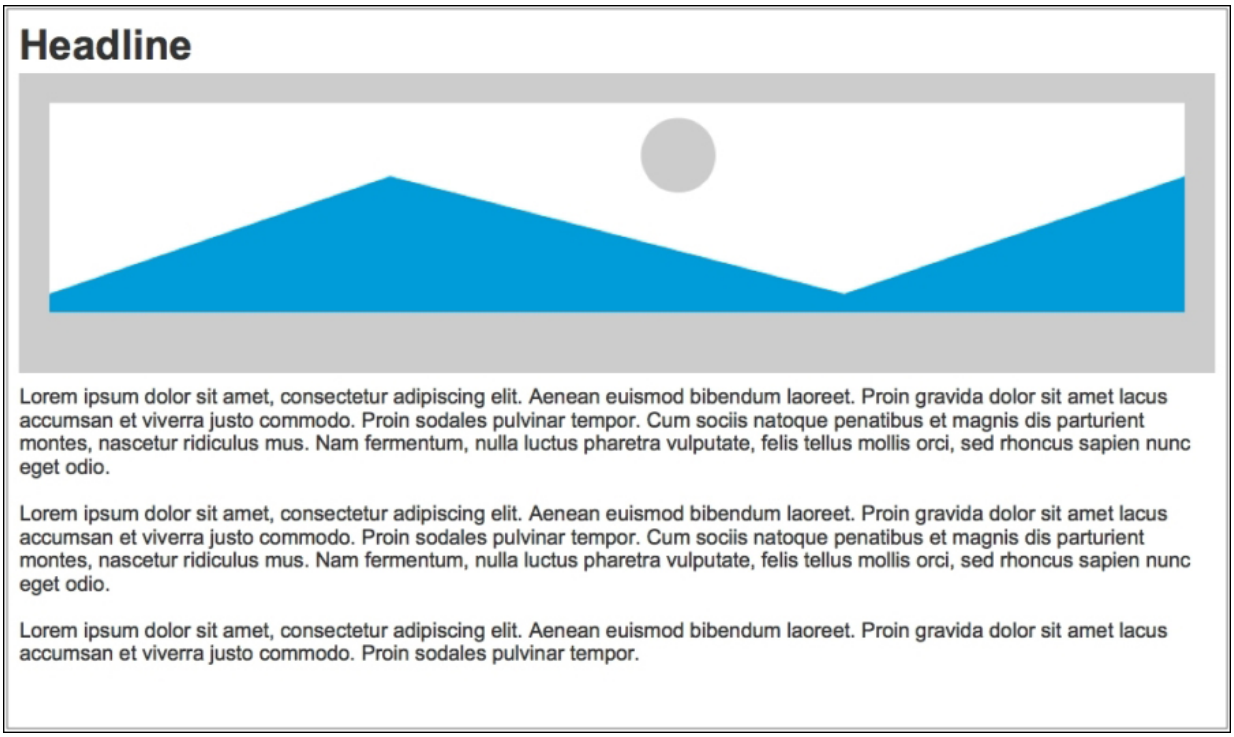
4. Under **Insert Variable or Function…**, type `[HorizontalOffset + LVAR1.length * 7]]`.
5. Beneath **Local Variables** under **Add Local Variable**, in the text field, type `LVAR1`.
6. In the first dropdown, **select value of variable**.

7. In the second drop down, select **BreadCrumbComponent**.
8. Click on **OK**.
9. Click on **OK**.

With the design and actions completed for the **BreadCrumb** master, we are now ready to assemble our **Secondary_Page_Body** master.

Assembling our Secondary_Page_Body master

We will construct a **Secondary_Page_Body** master that will be used on the **Journey**, **Topic Detail**, **Charter**, and **About Us** pages of our community site. Once completed, the **Secondary_Page_Body** master will look as follows:



The **Secondary_Page_Body** master will be made up of the following widgets:

Quantity	**Widget type**
1	Rectangle
1	H1 Heading 1

Quantity	Widget type
1	Image
1	Paragraph

We will now place widgets on the **Secondary_Page_Body** master.

Placing widgets on the Secondary_Page_Body master

All of the widgets used in the **Secondary_Page_Body** master can be found in the **Default** | **Common** dropdown in the widgets library. To place the widgets on the **Secondary_Page_Body** master, perform the following steps:

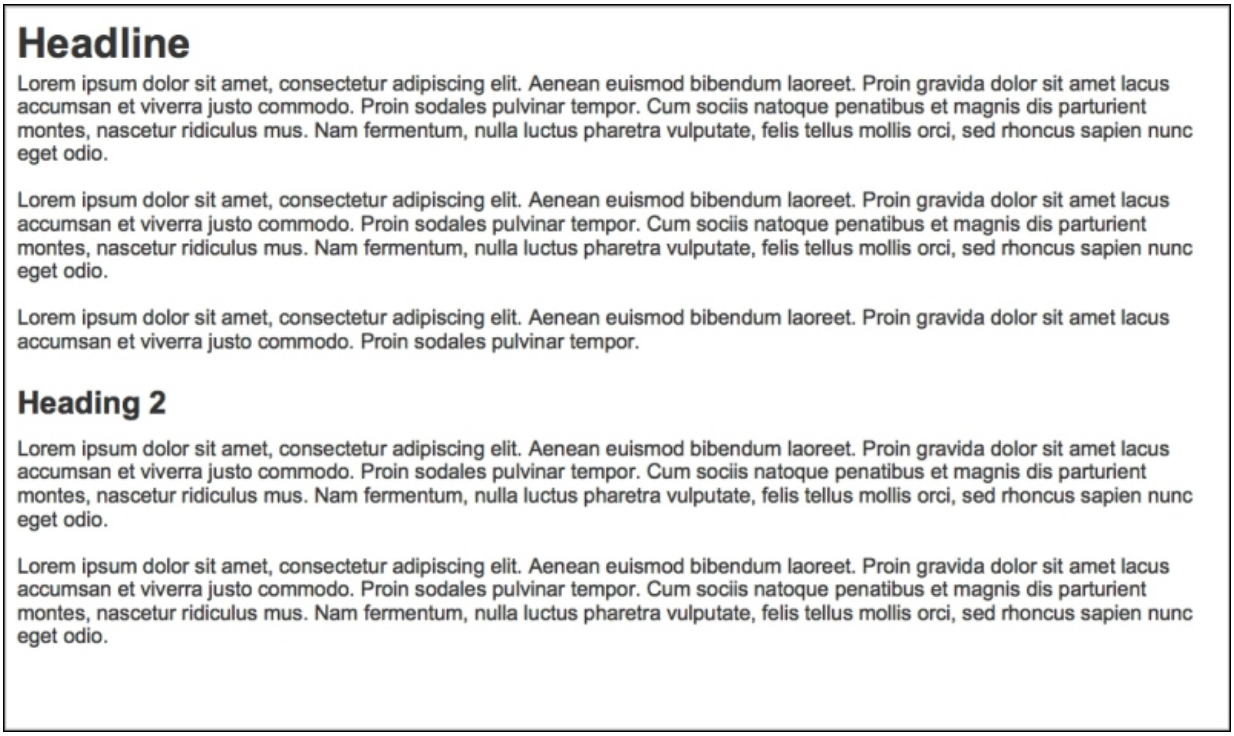
1. In the **Masters** pane, double-click on the icon next to the **Secondary_Page_Body** master to open it in the design area.
2. In the **Widgets** pane, drag the **Rectangle** widget and place at the coordinates (10,140). In the toolbar, change **w:** to `940` and **h:** to `555`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Background`.
3. In the **Widgets** pane, drag the **Heading 1** widget and place at the coordinates (20,150). Type `Headline`. In the toolbar, change **w:** to `134` and **h:** to `37`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Headline`.
4. In the **Widgets** pane, drag the **Image** widget and place at the coordinates (20,190). In the toolbar, change **w:** to `920` and **h:** to `230`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `HeroImage`.
5. In the **Widgets** pane, drag the **Paragraph** widget and place at the coordinates (20,430). In the toolbar, change **w:** to `920` and **h:**`240`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Copy`.

> ## Note
>
> The Paragraph widget displays Latin copy by default. Feel free to add or change the default copy to meet your requirements. To change the copy, click on the Paragraph widget at the coordinates (20,430) and enter the additional text.

With the design completed for the **Secondary_Page_Body** master, we are now ready to manufacture our **Informational_Page_Body** master.

Manufacturing our Informational_Page_Body master

We will manufacture an **Informational_Page_Body** master that will be used in the **Terms**, **Privacy**, and **Help** pages of our community site. Once completed, the **Informational_Page_Body** master will look as follows:



The **Informational_Page_Body** master will be made up of the following widgets:

Quantity	**Widget type**
1	Rectangle
1	H1 Heading 1
1	H2 Heading 2
2	Paragraph

We will now place the widgets on the **Informational_Page_Body** master.

Placing widgets on the Informational_Page_Body master

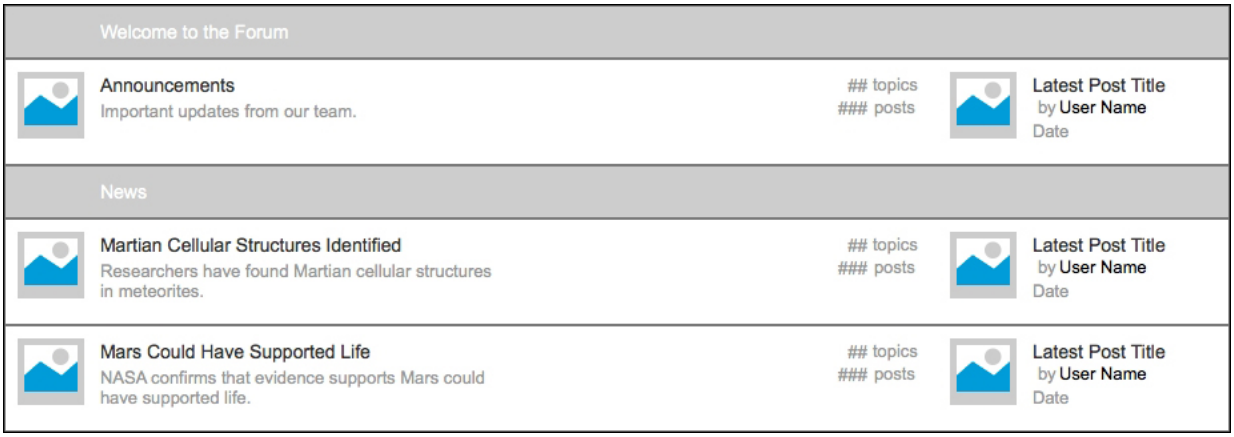
All of the widgets used in the **Informational_Page_Body** master can be found in the **Default** | **Common** dropdown in the widgets library. To place widgets on the **Informational_Page_Body** master, perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Informational_Page_Body** master to open it in the design area.
2. In the **Widgets** pane, drag the **Rectangle** widget and place at the coordinates (10,140). In the toolbar, change **w:** to `940` and **h:** to `555`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Background`.
3. In the **Widgets** pane, drag the **Heading 1** widget and place at the coordinates (20,150). Type `Headline`. In the toolbar, change **w:** to `134` and **h:**`37`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Headline`.
4. In the **Widgets** pane, drag the **Paragraph** widget, place it at the coordinates (20,190), and edit the text shown on the widget as required. In the toolbar, change **w:** to `920` and **h:** to `223`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Copy`.
5. In the **Widgets** pane, drag the **Heading 2** widget and place it at the coordinates (20,430). In the toolbar, change **w:** to `116` and **h:** to `28`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Heading2`.
6. In the **Widgets** pane, drag the **Paragraph** widget, place at the coordinates (20,470) and edit the text shown on the widget as appropriate. In the toolbar, change **w:** to `920` and **h:** to `180`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Copy2`.
7. With the design completed for the **Secondary_Page_Body** master, we are now ready to fabricate our **Forum** master.

Fabricating our Forum master

We will fabricate a **Forum** master that will be used on the **Forum** and **Topic List** pages of our community site. The **Forum** master enables a user to click on a category heading to view topics in the given category or click on an individual topic to view the topic detail page. We will use a single repeater that contains a dynamic panel with two states: **Topic** and **Category**. On the **OnItemLoad** event of the repeater, if the repeater item is a Category header, the **Category** state will be shown. If the repeater item contains a topic row, the **Topic** state will be shown.

Once completed, the **Forum** master will look as follows:



The **Forum** master will be made up of the following widgets:

Quantity	**Widget type**
1	Repeater
1	Dynamic Panel
2	Rectangle
11	Label
2	Hot Spot
2	Image

We will now place widgets on the **Forum** master.

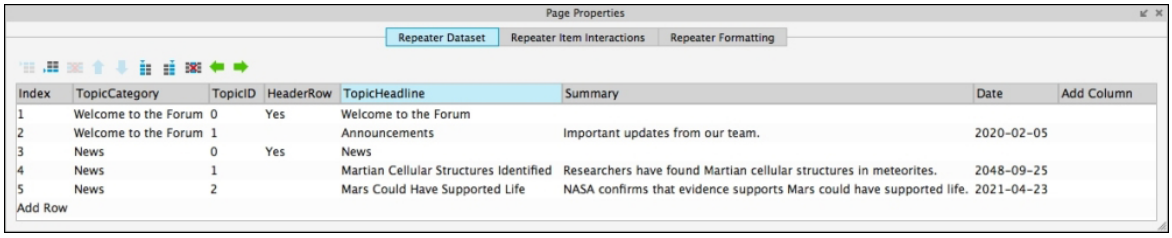
Formatting the Repeater and placing widgets on the Forum master

All of the widgets used in the **Forum** master can be found in the **Default** | **Common** dropdown of the widgets library. To place widgets on the **Forum** master, perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Forum** master to open in the design area.
2. In the **Widgets** pane, click on the **Repeater** widget. Drag the **Repeater** widget onto the design area and place at the coordinates (0,0). In the **Widget Interactions and Notes** pane, click on the **Repeater Name** field and type `ForumRepeater`.
3. Double-click on the **ForumRepeater** widget to open it in the design area.
4. Under the design area in the Repeater pane, click on the **Repeater Style** tab. To change the default Repeater style, perform the following steps:

    1. Navigate to the **Layout** dropdown.
    2. Click on the radio button next to **Horizontal**.
    3. Click on the checkbox next to the **Wrap (Grid)** field. In the **Items per row** field, enter `1`.

5. Now, click on the **Repeater Dataset** tab. The **ForumRepeater** will have a total of seven columns and five rows of data. Once completed, our **Repeater Dataset** tab will look as follows:



> ## Note
>
> Use the preceding screenshot as a reference to populate the Repeater dataset. Click on **Add Column** and **Add Row** to add additional columns and rows as needed.

6. To update and place widgets on the **Forum** master, click on the **Rectangle** widget at the coordinates (0,0). In the toolbar, change **w:** to `920` and **h:** to `80`. In the **Widget Interactions and Notes** pane, click in the **Shape Name** field and type `TopicBackground`.
7. Right-click on the **TopicBackground** widget at the coordinates (0,0) and click on **Convert to Dynamic Panel**. With the new **Dynamic Panel** widget selected,

perform the following steps:

1. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `ForumDP`.
2. In the **Widget Manager** pane, slow-double-click the **State1** state name and rename it to `Topic`.
3. In the **Widget Manager** pane, click on the state icon beside **Topic** two times, to open the **Topic** state in the design area.

8. In the **Widgets** pane, drag the **Label** widget and place at the coordinates (70,30).
9. Select the **Label** widget and type `This is a summary of the topic that supports two lines of text`. In the toolbar, change **w:** to `300` and **h:** to `30`.
10. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `TopicSummaryText`.
11. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Font** and perform the following steps:

    1. Click on the down arrow next to the Text Color icon. In the drop-down menu, in the **#** text field enter `999999`.
    2. Under the **Alignment + Padding** dropdown, click on the Align Left icon.

12. Repeat steps 8-11 to complete the design of the **Topic** state using the following table (* If applicable):

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)	Alignment + Padding*	Color* (In Hex #)
Image	(10,10)		`50`	`50`	`TopicThumbnailImg`		
Label	(70,10)	`Topic`	`460`	`20`	`TopicLabel`		
Label	(608,11)	`##`	`42`	`17`	`TopicCount`	Align Right	`999999`
Label	(609,28)	`###`	`41`	`17`	`PostsCount`	Align Right	`999999`

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)	Alignment + Padding*	Color* (In Hex #)
Label	(650,10)	topics	42	17	TopicsLabel		999999
Label	(651,27)	posts	38	17	PostsLabel		999999
Image	(710,10)		50	50	UserThumbnailImg		
Label	(770,10)	Latest Post Title	143	20	LatestPostLabel		
Label	(774,27)	by	22	17	byLabel		999999
Label	(790,27)	User Name	94	17	UserName		
Label	(770,45)	Date	100	15	PostDate		999999
Hot Spot	(10,10)		380	50	TopicHotSpot		

With the **Topic** state design completed for the **ForumDP**, we will now add and design the **Category** state. In the **Widget Manager** pane, under the **ForumDP**, right-click on the **Topic** state and in the menu, click on **Add State**. In the **Widget Manager** pane, slow-double-click the **State1** state name and rename it to `Category`. Click on the state

icon beside **Link** two times, to open the **Category** state in the design area. Perform the following steps:

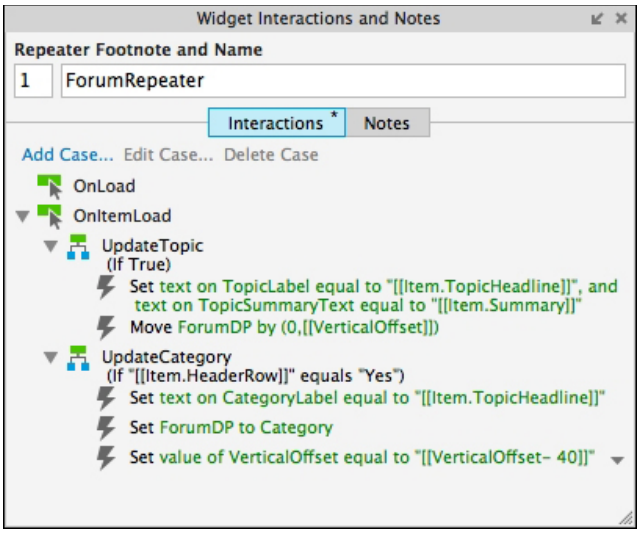
1. In the **Widgets** pane, drag the Rectangle widget and place at the coordinates (0,0).
2. In the toolbar, change **w:** to `920` and **h:** to `40`.
3. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `CategoryBackground`.
4. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Borders, Lines, + Fills** and perform the following steps:
    1. Click on the down arrow next to the Line Color icon. In the drop-down menu, in the **#** text field, enter `797979`.
    2. Click on the down arrow next to the Fill Color icon. In the drop-down menu, in the **#** text field, enter `CCCCCC`.
5. In the **Widgets** pane, drag the **Label** widget and place at the coordinates (70,10). Type `Category`. In the toolbar, change **w:** to `800` and **h:** to `20`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `CategoryLabel`.
6. In the **Widgets**pane, drag the **Hot Spot** widget and place at the coordinates (0,0). In the toolbar, change **w:** to `920` and **h:** to `40`. In the **Widget Interactions and Notes** pane, click on the **Hot Spot Name** field and type `CategoryHotSpot`.

With the design completed, we are now ready to update interactions for the Repeater item on the **Forum** repeater.

Updating the Repeater item's OnItemLoad event for the Forum repeater

The Repeater item for our **Forum** repeater contains the **ForumDP** dynamic panel with two states named as **Topic** and **Category**. A Hot Spot widget on each state provides a single **OnClick** interaction.

When the **OnItemLoad** event of the **Forum** repeater is triggered, there are two cases that will be evaluated: **UpdateTopic** and **UpdateCategory**. The following screenshot shows the actions that will be performed for each case:



When the repeater iterates, the **UpdateTopic** case will always be executed. As a result, the text will be updated in the **TopicLabel** and **TopicSummary** widgets with the corresponding values for the current repeater item. The **ForumDP** is then moved in the vertical direction based on **VerticalOffset**.

The **UpdateCategory** case will only be executed if the current **HeaderRow** repeater item value equals **Yes**. If this condition is met, text will be updated on the **CategoryLabel** widget with the corresponding value for the current repeater item. The **ForumDP** will be set to the **Category** state and the value of **VerticalOffset** will be updated.

We will now define the **OnItemLoad** event cases and actions for the **Forum** repeater.

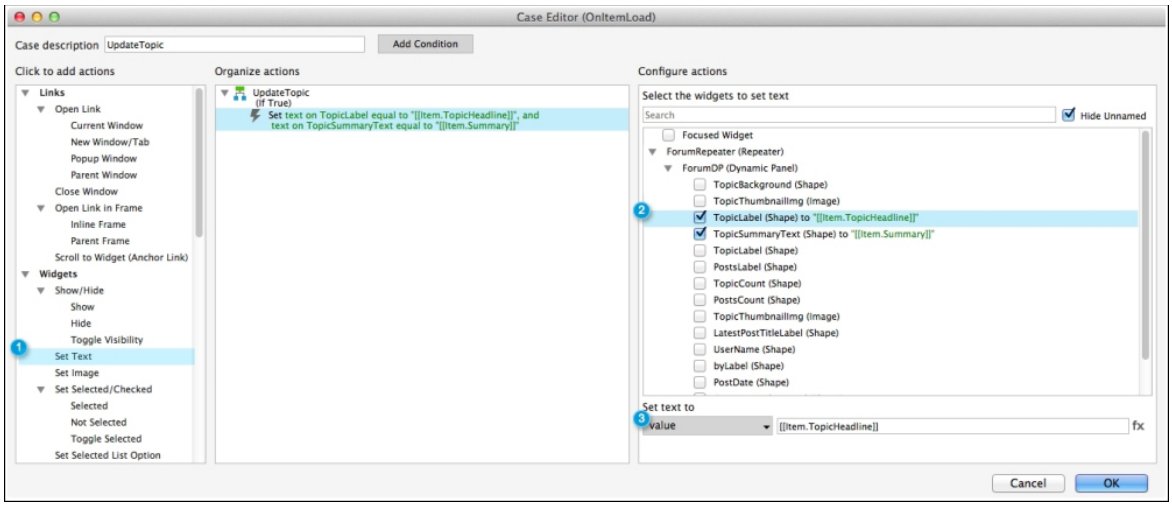
Realizing OnItemLoad event actions

In the **Masters** pane, double-click on the **Forum** master to open it in the design area. Click on the **ForumRepeater** at the coordinates (0,0). We will first define the **UpdateTopic** case. Then, we will define the **UpdateCategory** case.

Defining the UpdateTopic case

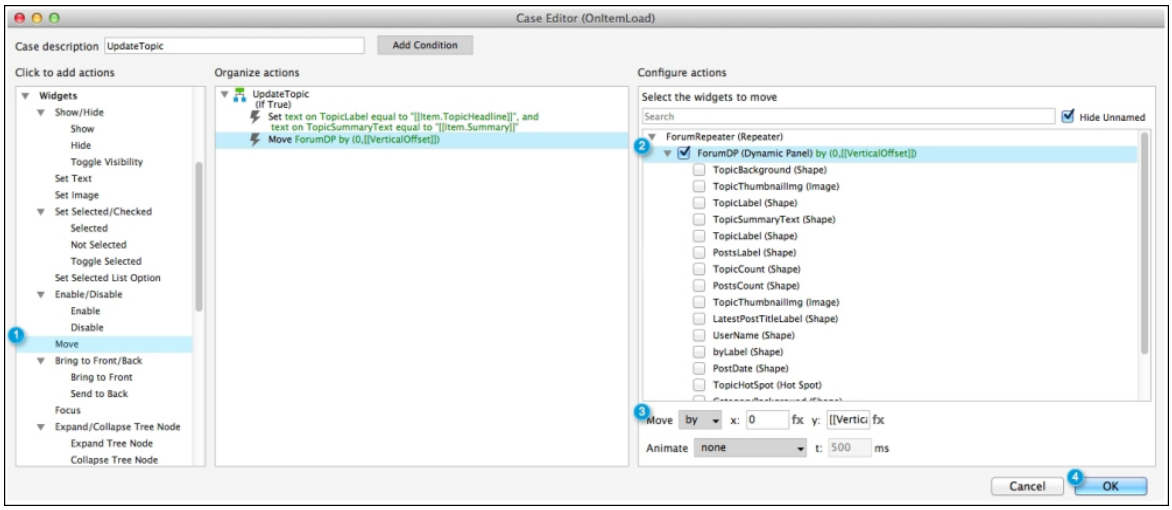
In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, then click on **More Events**, and double-click **OnItemLoad**. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

1. In the **Case description** field, type `UpdateTopic`.
2. Create the first action to set the value of the **TopicLabel** variable by performing the following steps:



1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
2. Under **Configure actions**, click the checkbox next to **TopicLabel**.
3. Under **Set text to**, click on the first dropdown and click **value**. In the text field, enter `[[Item.TopicHeadline]]`.
4. Repeat steps 2 and 3. For step 2, click on the checkbox next to **TopicSummaryText**. For step 3, in the text field enter `[[Item.Summary]]`.

3. Create the second action to move the **ForumDP** dynamic panel by following these steps:



1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Move**.
2. Under **Configure actions**, click on the checkbox next to **ForumDP**.
3. Under **Set variable to**, click on the first dropdown and click **by**. In the text field, enter `[[VerticalOffset]]`.
4. Click on **OK**.

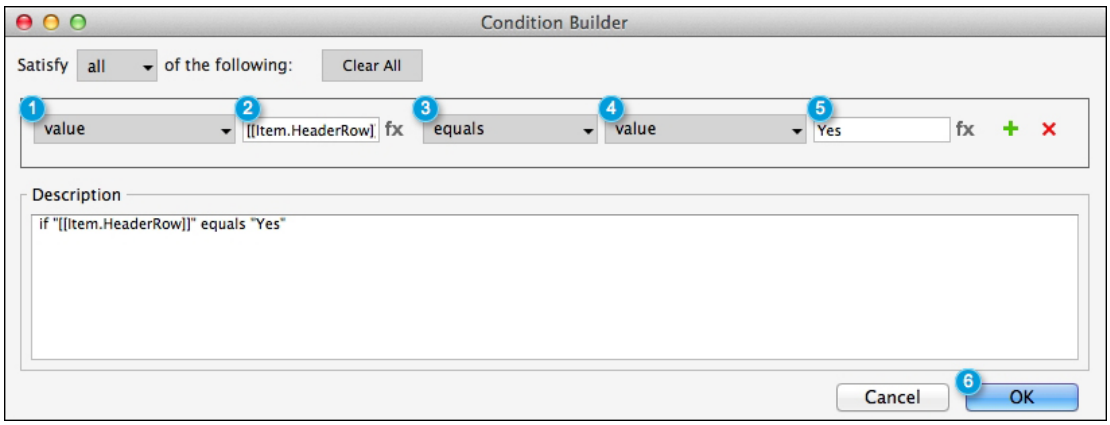
With the **UpdateTopic** case defined, we will now define the **UpdateCategory** case.

Defining the UpdateCategory case

In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, then click on **More Events**, and double-click on **OnItemLoad**. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

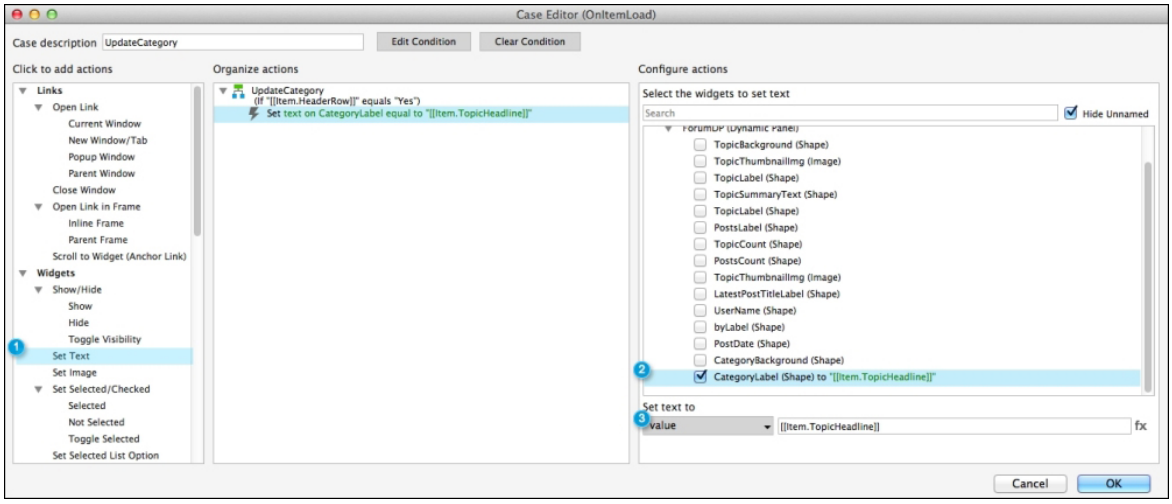
1. In the **Case description** field, type `UpdateCategory`.
2. Add the condition as follows:

    1. Click on the **Add Condition** button.
    2. In the **Condition Builder** dialog box, perform the following steps in the outlined condition box:



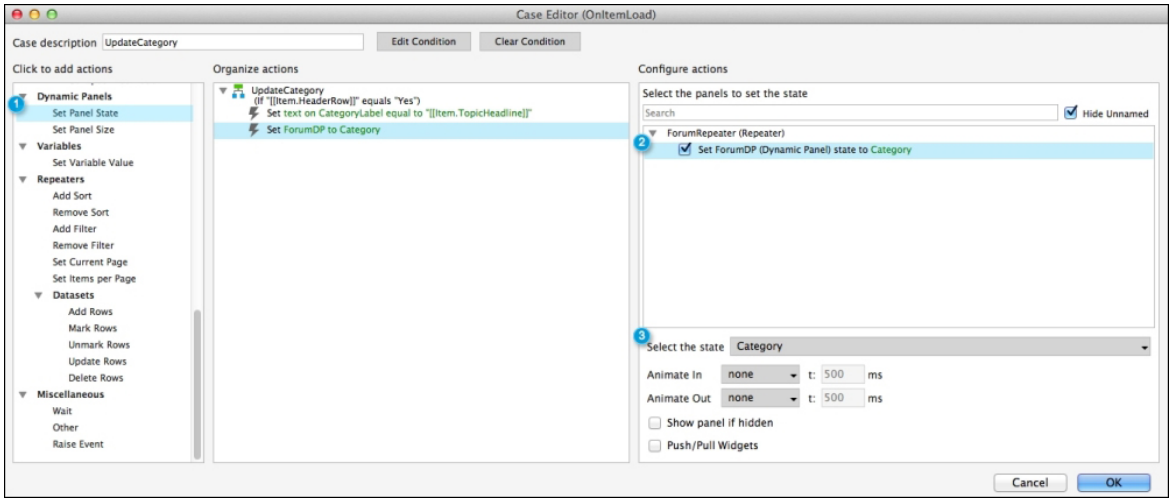
    3. In the first dropdown, select **value**.
    4. In the first text field, enter `[[Item.HeaderRow]]`.
    5. In the second dropdown, select **equals**.
    6. In the third dropdown, select **value**.
    7. In the second text field, enter `Yes`.
    8. Click on **OK**.

3. Now, create the first action. To set the value of the **CategoryLabel** variable, perform the following steps:



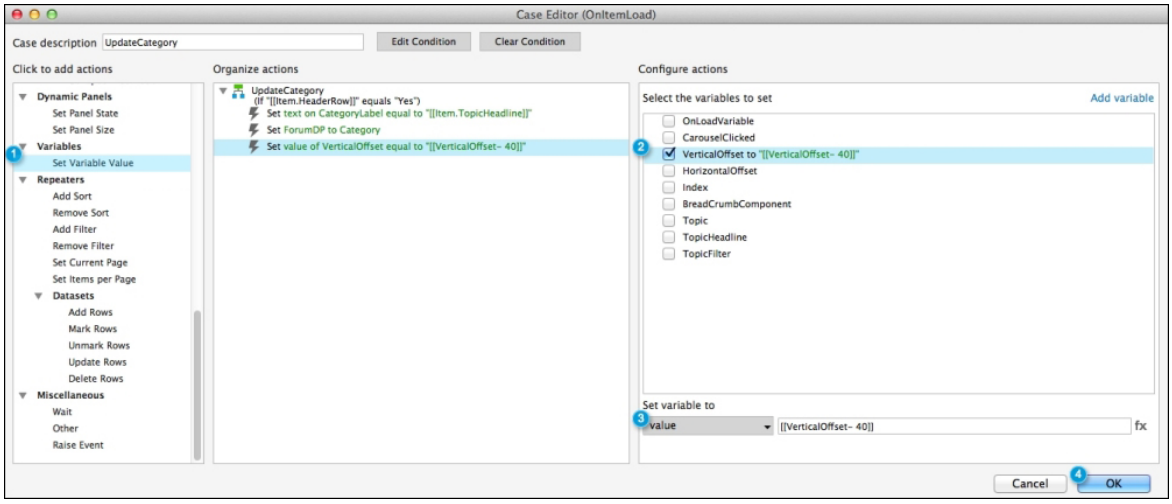
1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
2. Under **Configure actions**, click on the checkbox next to **CategoryLabel**.
3. Under **Set text to**, click the first dropdown and click on **value**. In the text field, enter `[[Item.TopicHeadline]]`.

4. Next, create the second action. To set the panel state for the **ForumDP** dynamic panel, perform the following steps:



1. Under **Click to add actions**, scroll to the **Dynamic Panels** drop-down menu and click on **Set Panel State**.
2. Under **Configure actions**, click on the checkbox next to **Set ForumDP**.
3. In **Select the state**, click the first dropdown and click on **Category**.

5. Now, create the third action. To set the value of the **VerticalOffset** variable, perform the following steps:



1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
2. Under **Configure actions**, click on the checkbox next to **VerticalOffset**.
3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `[[VerticleOffset - 40]]`.
4. Click on **OK**.
5. Under the **OnItemLoad** event, right-click on the **UpdateCategory** case and click on **Toggle IF/ELSE IF**.

With the **OnItemLoad** event cases and actions defined for the **Forum** repeater, we are now ready to build interactions for the **Forum** master.

Building interactions for the Forum master

Interactions with our **Forum** master occur as a result of the **OnClick** events. These **OnClick** events occur on the Hot Spot widgets as shown in the **Category** and **Topic** states of the **ForumDP** dynamic panel.

The **ForumDP** states, named widgets, and events are as follows:

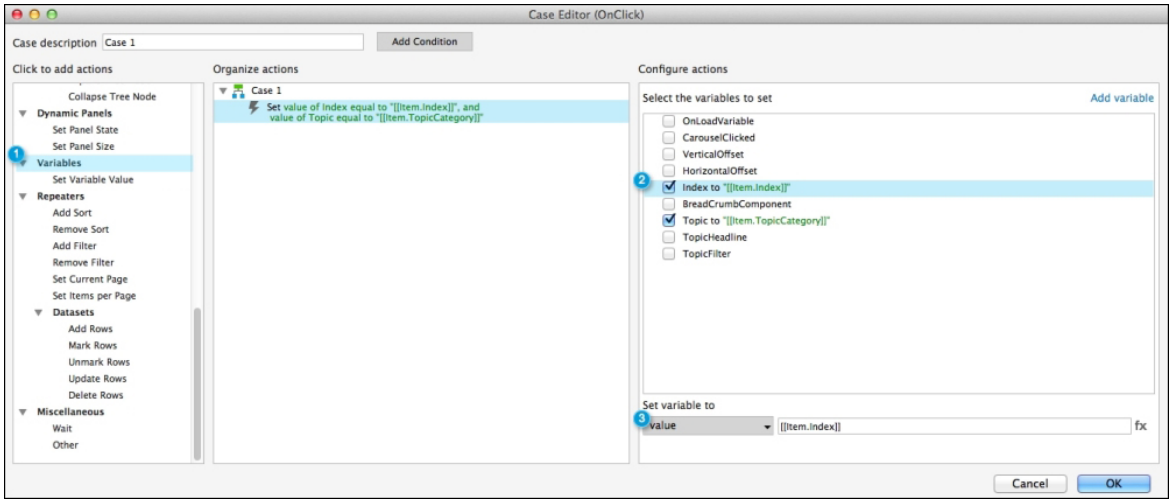
ForumDP state	**Named widget**	**Event(s)**
Category	**CategoryHotSpot**	**OnClick**
Topic	**TopicHotSpot**	**OnClick**

We will now enable the interactions for each Hot Spot starting with the **CategoryHotSpot**.

Enabling the OnClick interactions for the Category Hot Spot

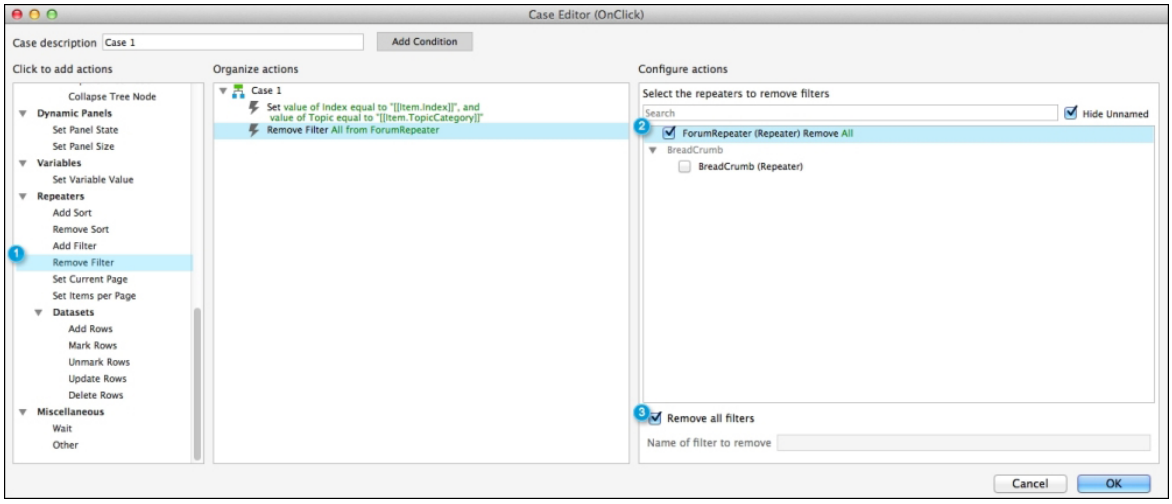
In the **Widget Manager** pane, under the **ForumDP** dynamic panel double-click on the **Category** state to open it in the design area. Click on the **CategoryHotSpot** at the coordinates (0,0). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click on **OnClick**. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

1. To create the first action, set the value of the **Index** and **Topic** variables and perform the following steps:



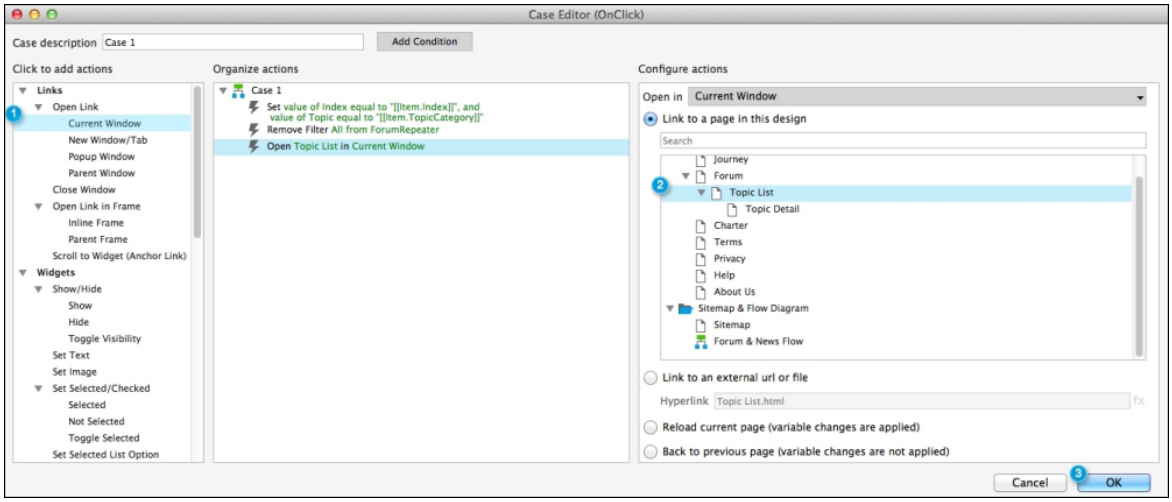
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **Index**.
    3. Under **Set text to**, click the first dropdown and click on **value**. In the text field, enter `[[Item.Index]]`.
    4. Repeat steps 2 and 3. For step 2, click on the checkbox next to **Topic**. For step 3, in the text field enter `[[Item.TopicCategory]]`.

2. Now, create the second action. To remove filter from the **ForumRepeater**, perform the following steps:



1. Under **Click to add actions**, scroll to the **Repeaters** drop-down menu and click on **Remove Filter**.
2. Under **Configure actions**, click on the checkbox next to **ForumRepeater**.
3. Click on the checkbox next to **Remove all filters**.

3. Next, create the third action. To open the **Topic List** page in the current window, perform the following steps:



1. Under **Click to add actions**, in the **Links** drop-down menu go under the **Open Link** drop-down menu and click on **Current Window**.
2. Under **Configure actions**, click on **Topic List**.
3. Click on **OK**.

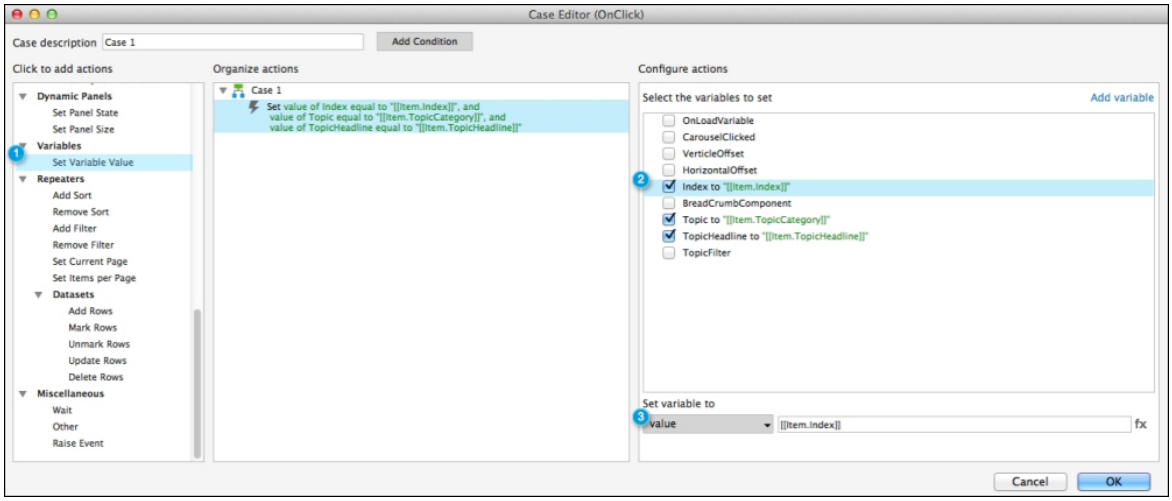
We are now ready to activate interactions for the **OnClick** event on the **Topic** Hot Spot.

Activating the OnClick interactions for the Topic Hot Spot

In the **Widget Manager** pane, under the **ForumDP** dynamic panel double-click on the **Topic** state to open it in the design area. Click on the **TopicHotSpot** at the coordinates (10,10). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and

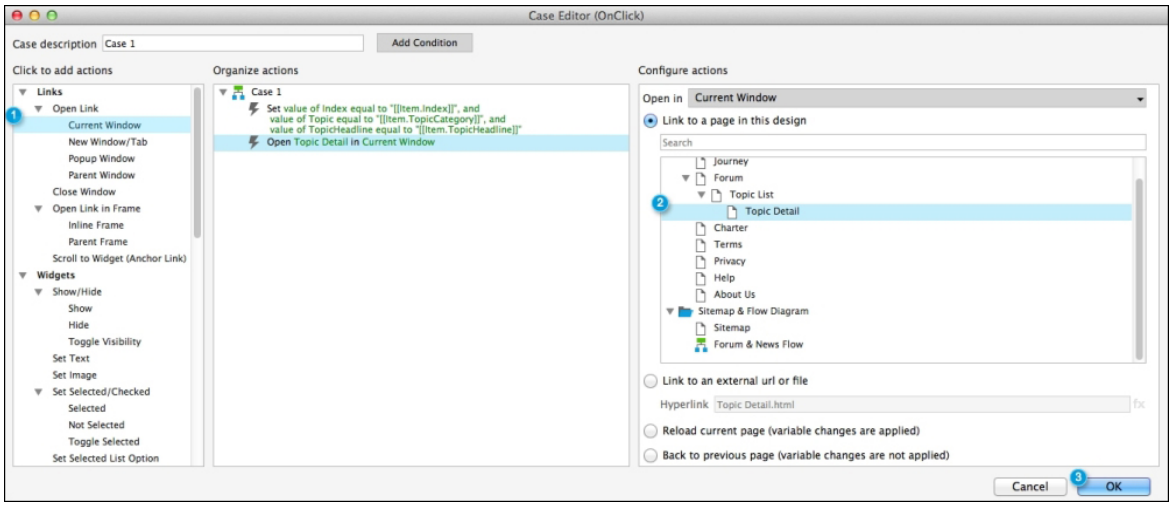
double-click **OnClick**. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

1. To create the first action, set the value of the **Index**, **Topic**, and **TopicHeadline** variables and perform the following steps:



   1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
   2. Under **Configure actions**, click on the checkbox next to **Index**.
   3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `[[Item.Index]]`.
   4. Repeat steps 2 and 3. For step 2, click the checkbox next to **Topic**. For step 3, in the text field enter `[[Item.TopicCategory]]`.
   5. Again, repeat steps 2 and 3. For step 2, click on the checkbox next to **TopicHeadline**. For step 3, in the text field enter `[[Item.TopicHeadline]]`.

2. Now, create the second action. To open the **Topic List** page in the current window, perform the following steps:

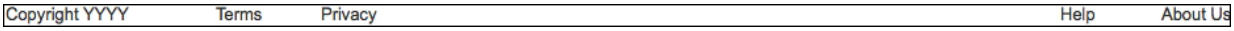


1. Under **Click to add actions**, in the **Links** drop-down menu go under the **Open Link** drop-down menu and click on **Current Window**.
2. Under **Configure actions**, click on **Topic Detail**.
3. Click on **OK**.

With the design and interactions completed for the **Forum** master, we are now ready to form our **Footer** master.

Forming our Footer master

Our **Footer** master will be used on all pages of our community site as identified in our sitemap. Once completed, the **Footer** master will look as follows:



The **Footer** master will be made up of five Label widgets . We are now ready to place widgets on the **Footer** master.

Placing widgets on the Footer master

The Label widget used in the **Footer** master can be found in the **Default** | **Common** dropdown in the widgets library. To place widgets on the **Footer** master, perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Footer** master to open in the design area.
2. In the **Widgets** pane, drag the **Label** widget and place at coordinates (10,715).
3. With the **Label** widget selected, perform the following steps:

    1. Type `Copyright YYYY`. In the toolbar, change **w:** to `140` and **h:** to `15`.
    2. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `CopyrightLink`.

4. Repeat steps 2 and 3 to complete the design of the **Footer** master using the following table:

Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
(170,715)	`Terms`	`60`	`15`	`TermsLink`
(250,715)	`Privacy`	`60`	`15`	`PrivacyLink`

Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
(813,715)	Help	27	15	HelpLink
(890,715)	About Us	60	15	AboutUsLink

With the design completed for the **Footer** master, we are now ready to define interactions.

Building interactions for our Footer master

The **Footer** master will have the **OnClick** events defined for the following:

- **TermsLink**
- **PrivacyLink**
- **HelpLink**
- **AboutUsLink**

We will now define the interactions for each label widget.

Defining interactions for our Label widgets

To define the interactions for the four Label widgets, perform the following steps:

1. To define the **OnClick** interaction, click on the **TermsLink** at the coordinates (170,715). In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Create Link…**. In the pop-up sitemap, click on **Terms**.
2. Repeat step 1 to complete the interactions for the **Footer** master using the following table:

Label widget	Coordinates	Create Link (to Page in Sitemap)
Terms	(170,715)	**Terms**
Privacy	(250,715)	**Privacy**

Label widget	**Coordinates**	**Create Link** **(to Page in Sitemap)**
Help	(813,715)	**Help**
About Us	(890,715)	**About Us**

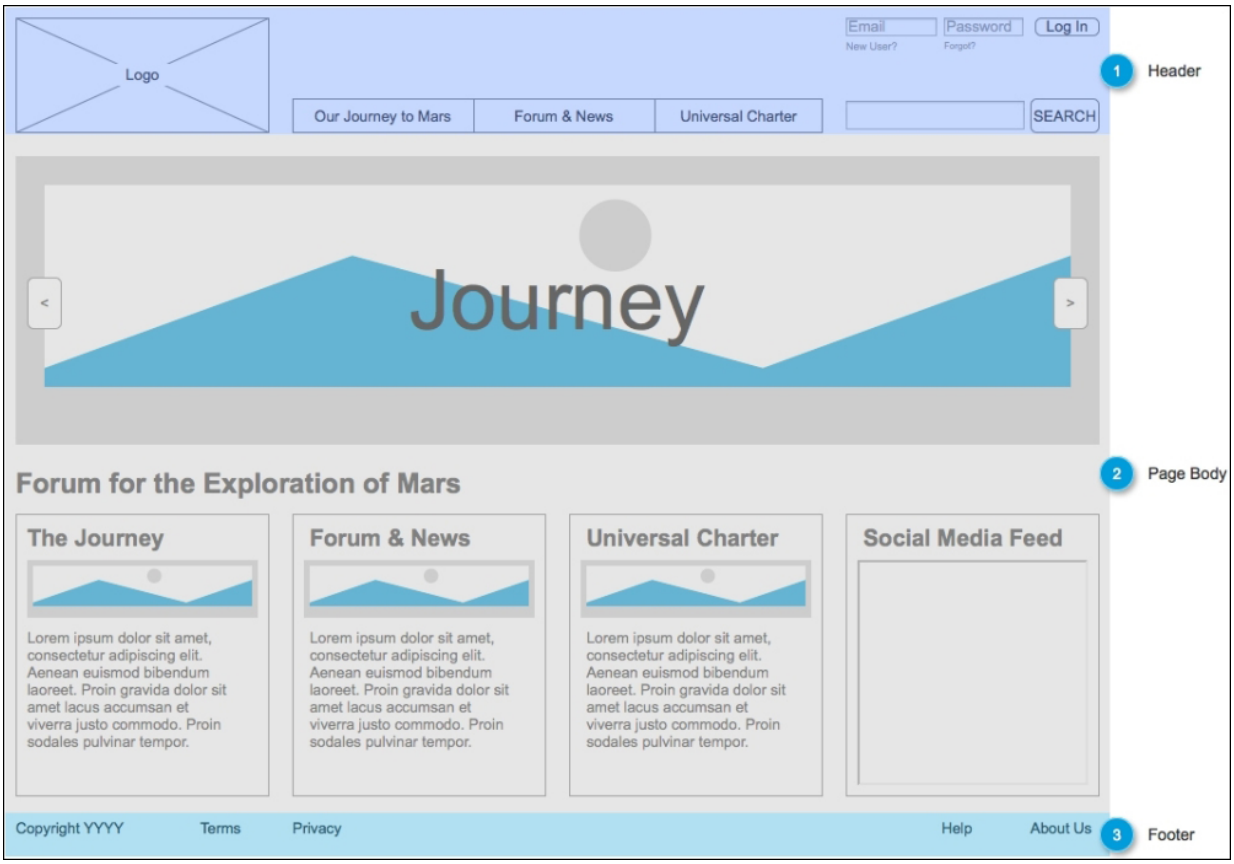
With the design and interactions completed for the **Footer** master, we are now ready to place the masters on each page of our community site.

Finalizing page design for our community site

To finalize the page design for our community site, we will first complete the design and interactions for the page body on the **Home** page. We then will place masters on each page of our community site. Finally, we will define the **OnPageLoad** events for a given page.

Placing widgets and masters on the Home page

Once completed, our **Home** page with the header master, page body, and footer master with each section highlighted will look as follows:



Before placing the header and footer masters, we will design the following elements for the page body:

- Carousel
- Heading and calls to action
- Social media feed

Next, we will design the page body for our home page.

Designing the page body

The page body will be comprised of the following widgets:

Quantity	**Widget type**
4	Hot Spot
1	

Quantity	Widget type
	Dynamic Panel
6	 Image
2	 Button Shape
1	H1 Heading 1
4	 Rectangle
4	H2 Heading 2
3	 Paragraph
1	 Inline Frame

We will start by placing the widgets and adding the interactions for the carousel.

Placing widgets and interactions for the dynamic carousel

The first widget that we will place is a hidden Hot Spot named **CheckForClick**. To make sure that the carousel stops the autorotation when the user clicks on the **Previous** or **Next** button, we set the focus event on the **CheckForClick** Hot Spot. The **CheckForClick** Hot Spot verifies that the value of **CarouselClicked** is `0`. If **CarouselClicked** is `0` the **Set Panel State** action for **CarouselDP** sets the dynamic

panel to the **Next** state. This is how we keep the carousel from advancing, if the user clicks on the **Previous** or **Next** button during a wait state.

Our carousel will have three states. Each state will have an Image widget with an **OnClick** event. If an image is clicked, the corresponding page will open in the current window.

All of the widgets used for the carousel can be found in the **Default** | **Common** dropdown in the widgets library. To place widgets on the **Home** page, perform the following steps:

1. In the **Masters** pane, double-click on the Page icon next to the **Home** page to open in the design area.
2. In the **Widgets** pane, drag the **Hot Spot** widget and place at the coordinates (95,35). In the toolbar, change **w:** to `50` and **h:** to `50`. In the **Widget Interactions and Notes** pane, click on the **Hot Spot Name** field and type `CheckForClick`.
3. In the **Widgets** pane, drag the **Image** widget and place at the coordinates (10,130). In the toolbar, change **w:** to `940` and **h:** to `250`. Right-click on the **Image** widget at the coordinates (10,130) and click on **Convert to Dynamic Panel**. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `CarouselDP`.
4. In the **Widget Manager** pane, right-click the State icon beside **State1** and click on **Duplicate State** to create `State2`.
5. Repeat step 4 to create `State3`.
6. In the **Widgets** pane, drag the **Button Shape** widget and place at the coordinates (20,235). Type `<`. In the toolbar, change **w:** to `30` and **h:** to `45`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `PreviousButton`.
7. In the **Widgets** pane, drag the **Button Shape** widget and place at the coordinates (910,235). Type `>`. In the toolbar, change **w:** to `30` and **h:** to `45`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `NextButton`.
8. In the **Widget Manager** pane, click the State icon beside **State1** two times to open **State1** in the design area. Click on the Image widget at the coordinates (10,130). Type `Journey`. In the **Widget Interactions and Notes** pane, click on the **Image Name** field and type `Image1`.
9. Repeat step 6 to complete the design of the **CarouselDP** for **State2** and **State3** using the following table:

CarouselDP state	Text* (Shown on Widget)	Name field (In the Widget Interactions and Notes pane)
State2	`Latest News`	`Image2`
State3	`UniversalCharter`	`Image3`

With the design completed, we are now ready to build the interactions for **CarouselDP**.

Building interactions for the carousel dynamic panel

We will first enable the interactions for the **CheckForClick** Hot Spot. Then, we will define the **OnClick** interactions for the states of the **CarouselDP** dynamic panel. Finally, we will activate the **Previous** and **Next** button interactions.

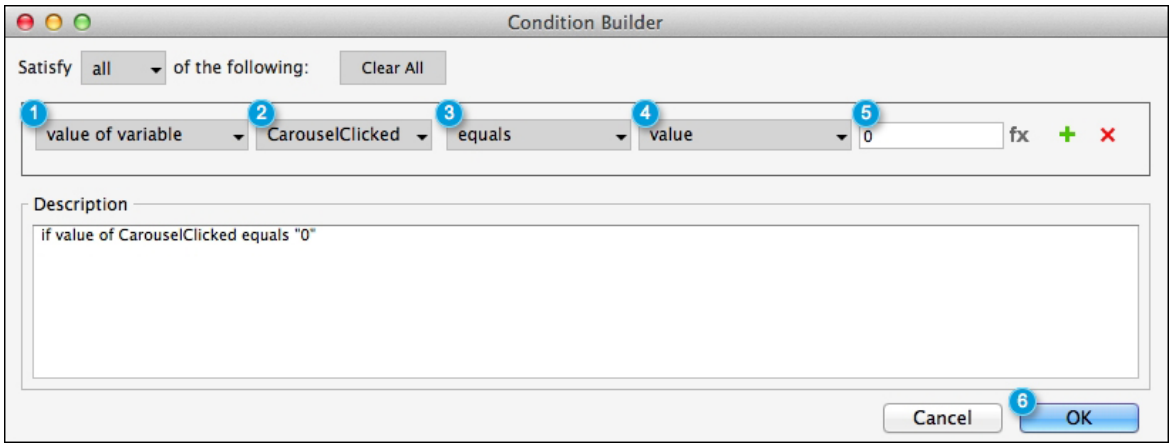
Enabling interactions for the CheckForClick Hot Spot

The **CheckForClick** Hot Spot on the **OnFocus** event will set the panel state of **CarouselDP** to Next, if **CarouselClicked** is equal to `0`. Enable this interaction by performing the following steps:

1. To activate the **OnFocus** event, click on the **CheckForClick** Hot Spot at the coordinates (95,35).
2. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, then click on **More Events**, and double-click **OnFocus**.

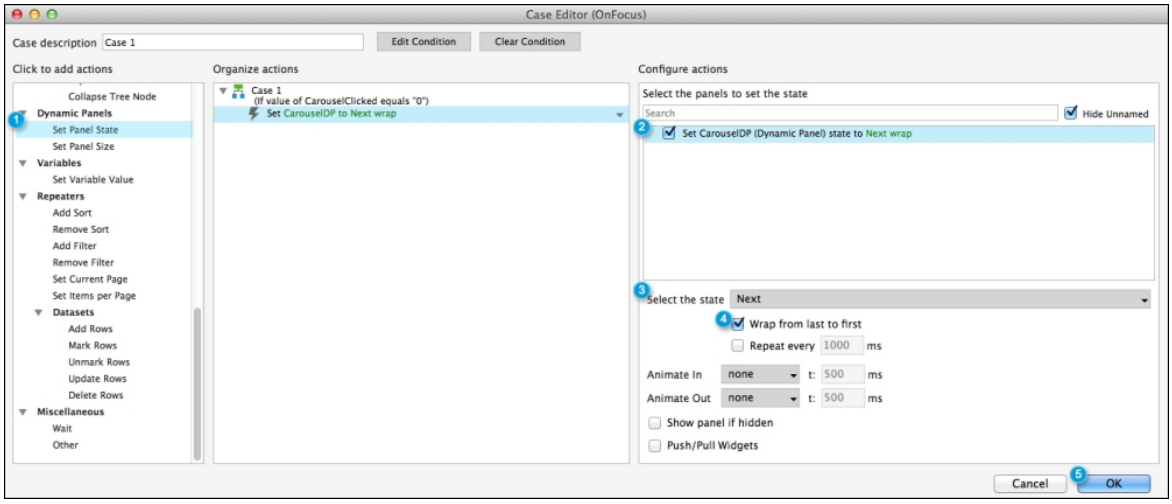
   A **Case Editor** dialog box will open.

3. Now to add the new condition, click on the **Add Condition** button.
4. You will see the **Condition Builder** dialog box; in the outlined condition box as shown in the screenshot, perform the following steps:



1. In the first dropdown, select **value of variable**.
2. In the second dropdown, select **CarouselClicked**.
3. In the third dropdown, select **equals**.
4. In the fourth dropdown, select **value**.
5. In the text field, enter `0`.
6. Click on **OK**.

5. Let's now create the action. To set the panel state of the **CarouselDP** to **Next**, perform the following steps:



1. Under **Click to add actions**, scroll to the **Dynamic Panels** drop-down menu and click on **Set Panel State**.
2. Under **Configure actions**, click the checkbox next to **CarouselDP**.
3. Click on the **Select the state** dropdown and select **Next**.
4. Click on the checkbox next to **Wrap from last to first**.
5. Click on **OK**.

Next, we will define the **OnClick** interactions for the states of the **CarouselDP** dynamic panel.

Defining the OnClick interactions for the Carousel dynamic panel

To define the **OnClick** interactions for each of the three states of the **CarouselDP** dynamic panel, perform the following steps:

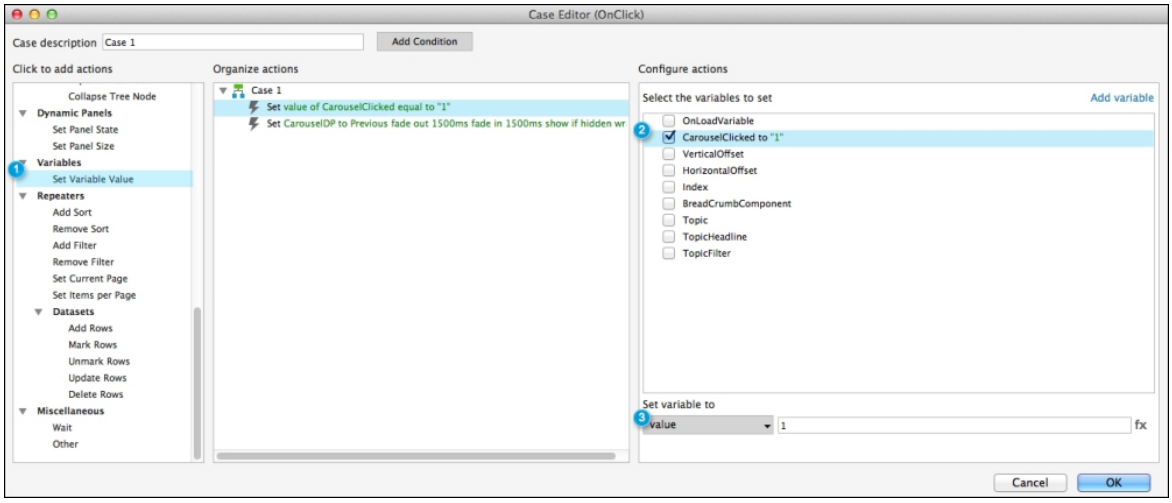
1. In the **Widget Manager** pane, click on the State icon beside **State1** two times to open the state in the design area. Click on the **Image** widget at the coordinates (0,0). In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Create Link…**. In the pop-up sitemap, click on **Journey**.
2. In the **Widget Manager** pane, click on the State icon beside **State2** two times to open the state in the design area. Click on the **Image** widget at the coordinates (0,0). In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Create Link…**. In the pop-up sitemap, click on **Forum**.
3. In the **Widget Manager** pane, click the State icon beside **State1** two times to open the state in the design area. Click on the **Image** widget at the coordinates

(0,0). In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Create Link…**. In the pop-up sitemap, click on **Charter**.

Next, we will activate the interactions for the **Previous** and **Next** buttons.

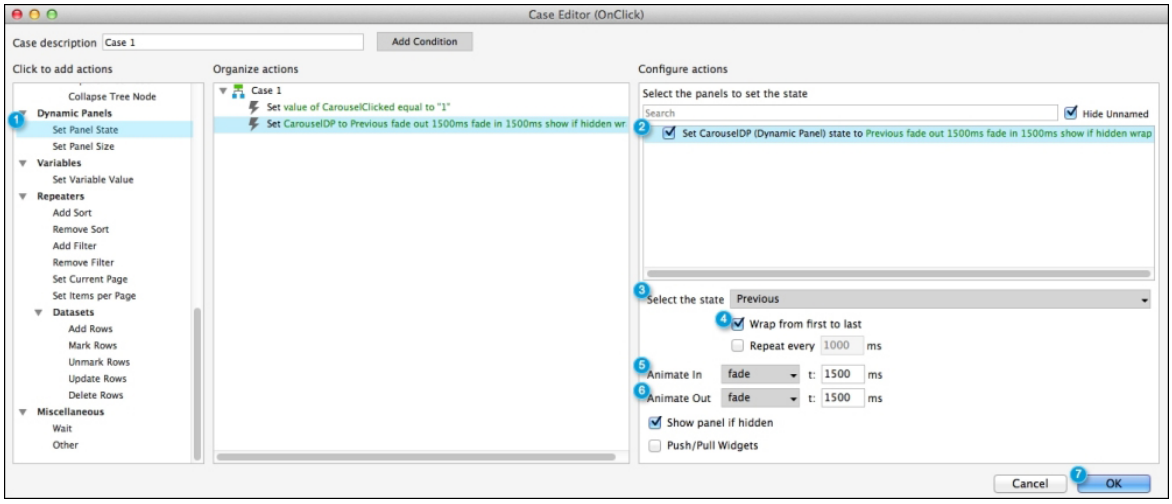
Activating the Previous and Next buttons

1. To activate the **OnClick** event for **Previous**, click on the **Previous** button at the coordinates (20,235).
2. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, then click on **Add case…**. A **Case Editor** dialog box will open.
3. Now, create the first action. To set the value of the **CarouselClicked** variable, perform the following steps:



   1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
   2. Under **Configure actions**, click on the checkbox next to **CarouselClicked**.
   3. Under **Set variable to**, click the first dropdown and click on **value**. In the text field, enter `1`.

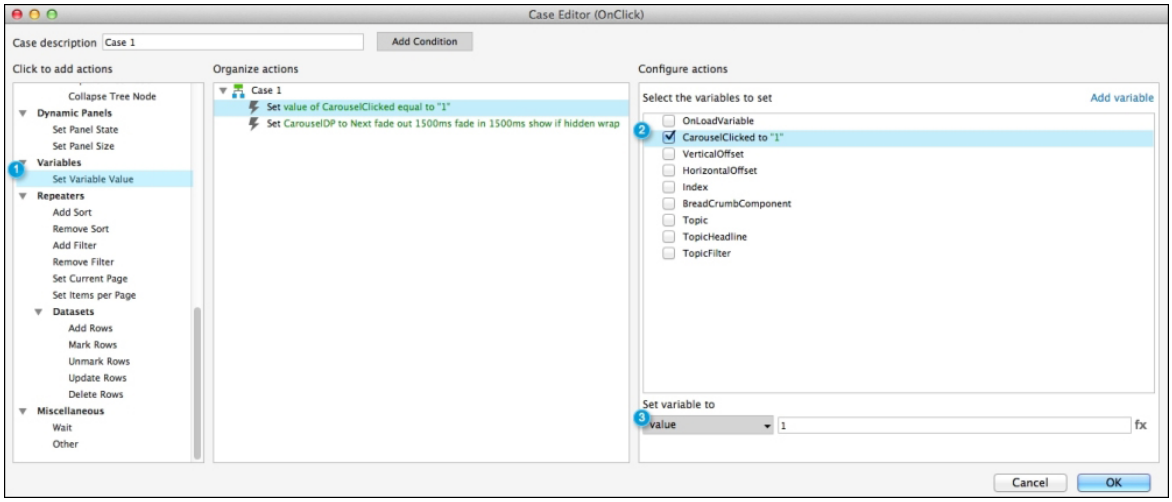
4. Create the second action. To set panel state of the **CarouselDP** to **Previous**, perform the following steps:



1. Under **Click to add actions**, scroll to the **Dynamic Panels** drop-down menu and click on **Set Panel State**.
2. Under **Configure actions**, click on the checkbox next to **CarouselDP**.
3. Click on the **Select the state** dropdown and select **Previous**.
4. Click on the checkbox next to **Wrap from last to first**.
5. Click on the **Animate In** dropdown and select **fade**. In the text field beside **t:**, type `1500`.
6. Click on **Animate Out** dropdown and select **fade**. In the text field beside **t:**, type `1500`.
7. Click on **OK**.

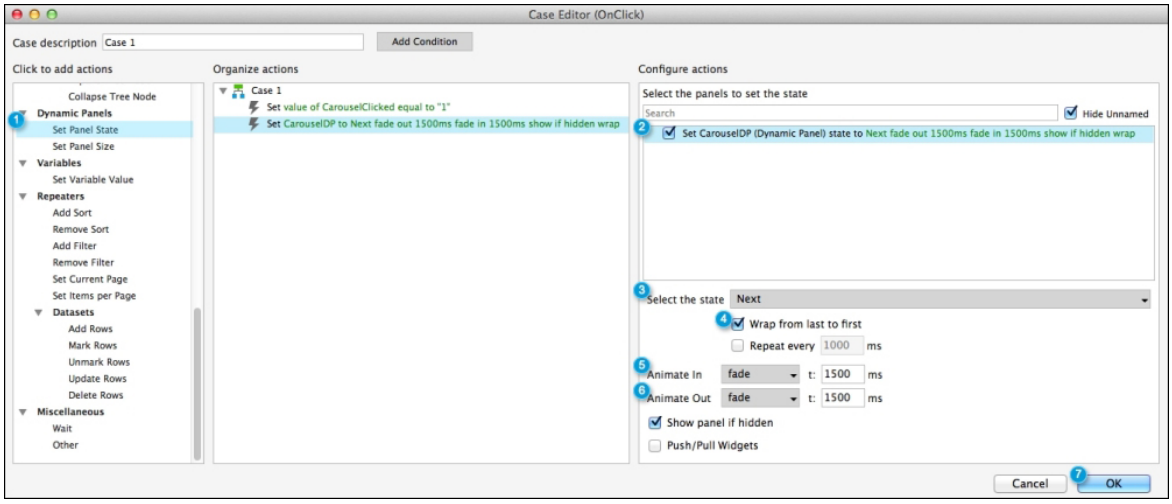
To activate the **OnClick** event for **Next**, click on the **Next** button at the coordinates (910,235). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, and then click on **Add case…**. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. To set the value of the **CarouselClicked** variable, perform the following steps:



1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
2. Under **Configure actions**, click on the checkbox next to **CarouselClicked**.
3. Under **Set variable to**, click on the first dropdown and click **value**. In the text field, enter `1`.

2. Now, create the second action. To set the **Panel** state of the **CarouselDP** dynamic panel to **Next**, perform the following steps:



1. Under **Click to add actions**, scroll to the **Dynamic Panels** drop-down menu and click on **Set Panel State**.
2. Under **Configure actions**, click on the checkbox next to **CarouselDP**.
3. Click on the **Select the state** dropdown and select **Next**.
4. Click the checkbox next to **Wrap from last to first**.
5. Click on the **Animate In** dropdown and select **fade**. In the text field beside **t:**, type `1500`.
6. Click on the **Animate Out** dropdown and select **fade**. In the text field beside **t:**, type `1500`.
7. Click on **OK**.

With the design and interactions for the carousel dynamic panel completed, we are now ready to place the widgets and add the interactions for the heading and calls to action.

Placing widgets and adding interactions for calls to action

The first widget we will place is the **Heading 1** widget. We will then construct three calls-to-action boxes. To place the Heading 1 widget and create the calls-to-action boxes, perform the following steps:

1. In the **Masters** pane, double-click on the Page icon next to the **Home** page to open in the design area.
2. In the **Widgets** pane, drag the **Heading 1** widget H1 and place at the coordinates (10,400). Type `Forum for the Exploration of Mars`. In the toolbar, change **w:** to `388` and **h:** to `28`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Heading1`.

3. In the **Widgets** pane, drag the **Rectangle** widget  and place at the coordinates (10,440). In the toolbar, change **w:** to `220` and **h:** to `245`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section1_Background`.
4. In the **Widgets** pane, drag the **Heading 2** widget  and place at the coordinates (20,450). Type `The Journey`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section1_Heading`.
5. In the **Widgets** pane, drag the **Image** widget  and place at the coordinates (20, 480). In the toolbar, change **w:** to `200` and **h:** to `50`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section1_Image`.
6. In the **Widgets** pane, drag the **Paragraph** widget  and place at the coordinates (20,540). In the toolbar, change **w:** to `180` and **h:** to `117`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section1_Copy`. Adjust **Paragraph** text to fit within the **Section1_Background** rectangle.
7. In the **Widgets**pane, drag the **Hot Spot** widget  and place at the coordinates (10,440). In the toolbar, change **w:** to `220` and **h:** to `245`. In the **Widget Interactions and Notes** pane, click on the **Hot Spot Name** field and type `Journey_HotSpot`.
8. In the **Widgets** pane, drag the **Rectangle** widget  and place at the coordinates (250,440). In the toolbar, change **w:** to `220` and **h:** to `245`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section2_Background`.
9. In the **Widgets** pane, drag the **Heading 2** widget  and place at the coordinates (265,450). Type `Forum & News`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section2_Heading`.
10. In the **Widgets** pane, drag the **Image** widget  and place at the coordinates (260,480). In the toolbar, change **w:** to `200` and **h:** to `50`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section2_Image`.
11. In the **Widgets** pane, drag the **Paragraph** widget  and place at the coordinates (265,540). In the toolbar, change **w:** to `180` and **h:** to `117`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section2_Copy`. Adjust **Paragraph** text to fit within the **Section1_Background** rectangle.
12. In the **Widgets** pane, drag the **Hot Spot** widget  and place at the coordinates (250,440). In the toolbar, change **w:** to `220` and **h:** to `245`. In the **Widget Interactions and Notes** pane, click on the **Hot Spot Name** field and type `News_HotSpot`.
13. In the **Widgets**pane, drag the **Rectangle** widget  and place at the coordinates (490,440). In the toolbar, change **w:** to `220` and **h:** to `245`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section3_Background`.
14. In the**Widgets** pane, drag the **Heading 2** widget  and place at the coordinates (505,450). Type `Universal Charter`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section3_Heading`.
15. In the **Widgets** pane, drag the **Image** widget  and place at the coordinates (500, 480). In the toolbar, change **w:** `200` and **h:** `50`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Section3_Image`.
16. In the **Widgets** pane, drag the **Paragraph** widget  and place at the coordinates (505,540). In the toolbar, change **w:** to `180` and **h:** to `117`. In the **Widget Interactions**

and Notes pane, click on the **Shape Name** field and type `Section3_Copy`. Adjust **Paragraph** text to fit within the **Section1_Background** rectangle.

17. In the **Widgets** pane, drag the **Hot Spot** widget and place at the coordinates (490,440). In the toolbar, change **w:** to `220` and **h:** to `245`. In the **Widget Interactions and Notes** pane, click on the **Hot Spot Name** field and type `Charter_HotSpot`.

To add the **OnClick** interaction for each of the three Hot Spots, perform the following steps:

1. Click on the **Hot Spot** widget at the coordinates (10,440). In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Create Link…**. In the pop-up sitemap, click on **Journey**.
2. Click on the **Hot Spot** widget at the coordinates (250,440). In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Create Link…**. In the pop-p sitemap, click on **Forum**.
3. Click on the **Hot Spot** widget at the coordinates (490,440). In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Create Link…**. In the pop-up sitemap, click on **Charter**.

We will now add a social media feed to the home page.

Adding a social media feed

We will use an Inline Frame widget with the social aggregator Tint (http://www.tintup.com). After creating an account and adding a Tint (that is, social media feed), a URL will be provided for our feed. To add the social media feed, perform the following steps:

1. In the **Widgets** pane, drag the **Rectangle** widget and place at the coordinates (730,440). In the toolbar, change **w:** to `220` and **h:** to `245`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `SocialMediaFeedBackground`.
2. In the **Widgets** pane, drag the **Heading 2** widget H2 and place at the coordinates (745,450). Type `Social Media Feed`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `SocialMediaFeedHeading`.
3. In the **Widgets** pane, click on the **Inline Frame** widget. Drag the **Inline Frame** widget and place at the coordinates (740,480). Perform the following steps:
    1. In the **Widget Interactions and Notes** pane, click on the **Inline Frame Name** field and type `SocialMediaFeedIF`.
    2. In the toolbar, change **w:** to `200` and **h:** to `195`.
    3. Right-click on the **SocialMediaFeed Inline Frame** widget, click on **Scrollbars**, and click on **Never Show Scrollbars**.
    4. Right-click on the **SocialMediaFeed Inline Frame** widget and click on **Frame Target**. In the **Link Properties** dialog box, click on the radio button next to **Link to an external url or file**. In the **Hyperlink** field, paste the embed link to your Tint (for example, `http://www.tintup.com/axuredemo`).

With the page body completed for the home page, we are now ready to add the master and define actions for the **OnPageLoad** event.

Adding masters and defining actions for the home page

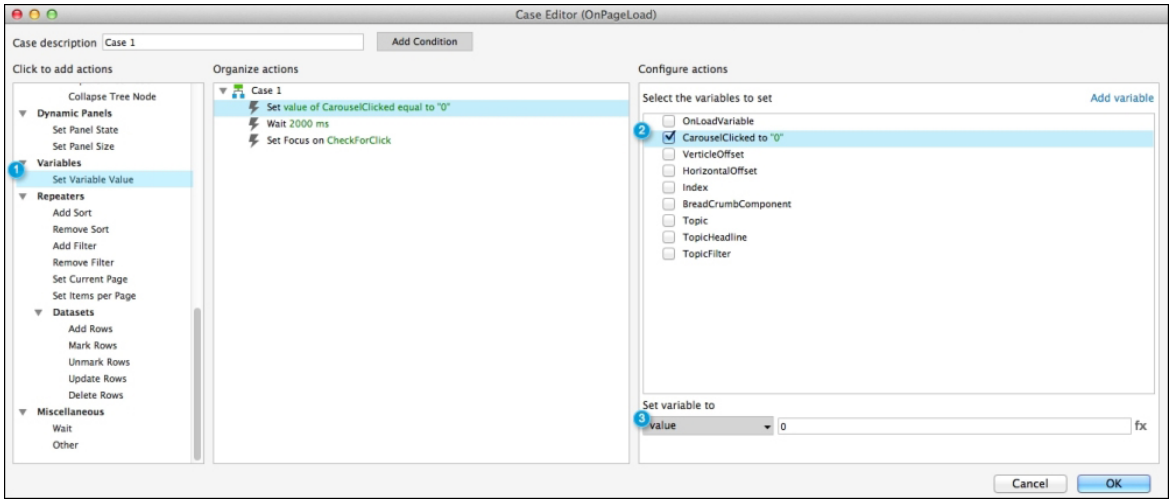
We will first add the **Header** and **Footer** masters to the home page. We will then define the actions for the **OnPageLoad** event. With the home page opened in the design area, perform the following steps:

1. In the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
2. In the **Masters** pane, drag the **Footer** master and drop it anywhere in the design area.

> ### Note
>
> The icon next to the **Header** and **Footer** master indicates that the drop behavior is locked to the master location. The masters with this drop behavior will automatically be placed in the design area at the same location as the corresponding master's location.

To define the actions for the **OnPageLoad** event, go to the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on **Add Case…**, so that a **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

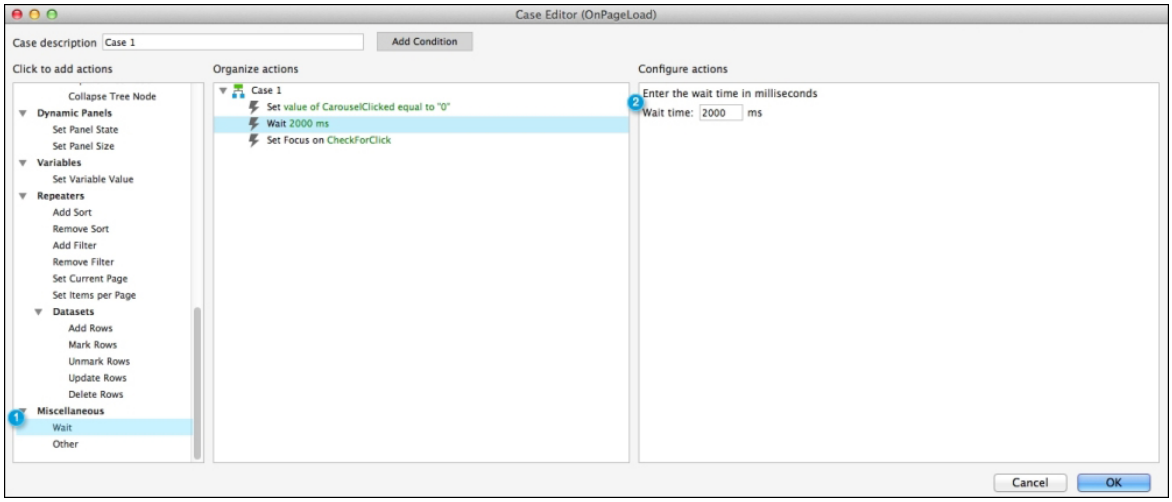
1. Create the first action. To set the value of the **CarouselClicked** variable, perform the following steps:



   1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
   2. Under **Configure actions**, click the checkbox next to **CarouselClicked**.

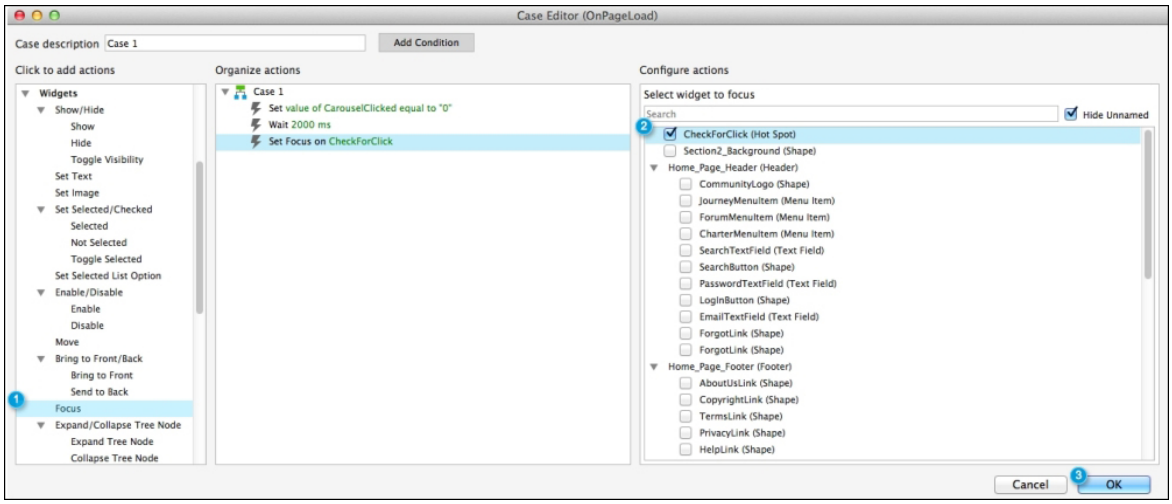
3. Under **Set variable to**, click the first dropdown and click on **value**. In the text field, enter `0`.

2. Create the second action to wait for 2000 ms by performing the following steps:



1. Under **Click to add actions**, scroll to the **Miscellaneous** drop-down menu and click on **Wait**.
2. Under **Configure actions**, beside **Wait time** in the text field, enter `2000`.

3. Create the third action to set focus on **CheckForClick** by performing the following steps:



1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Focus**.
2. Under **Configure actions**, click on the checkbox next to **CheckForClick**.
3. Click on **OK**.

With the home page design and interactions completed, we are now ready to complete the **Journey**, **Topic Detail**, **Charter**, and **About Us** pages.

Completing the design and interactions for the Journey, Topic Detail, Charter, and About Us pages

The **Journey**, **Topic Detail**, **Charter**, and **About Us** pages use the **Header**, **Secondary_Page_Body**, and **Footer** masters. To complete the design for these pages, perform the following steps:

1. In the**Masters** pane, double-click on the Page icon next to the **Journey** page to open in the design area.
2. In the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. In the **Masters** pane, drag the **Secondary_Page_Body** master and drop it anywhere in the design area.
4. In the **Masters** pane, drag the **Footer** master and drop it anywhere in the design area.
5. Repeat steps 1 to 4 to complete the designs for the **Topic Detail**, **Charter**, and **About Us** pages using the values given in the following table:

Page to Open **(Step 1)**
Topic Detail
Charter
About Us

Next, we will define the interactions for the **Journey**, **Topic Detail**, **Charter**, and **About Us** pages.

Defining interactions for the Journey, Topic Detail, Charter, and About Us pages

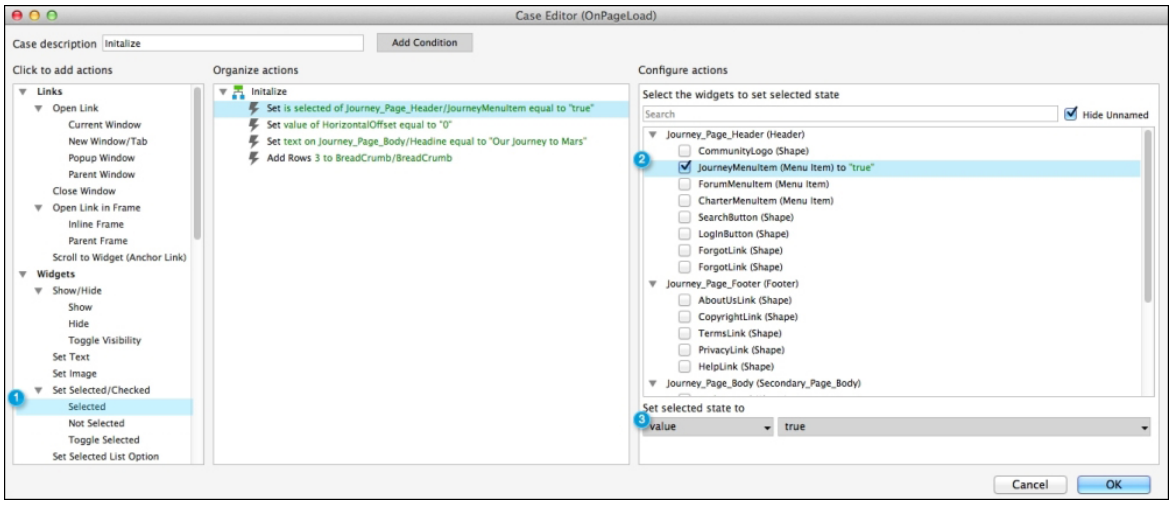
For these pages, we will first update the menu item and **BreadCrumb** to give the user feedback regarding which page they are currently on. We accomplish this by completing the **Set Selected** action by setting it to **True** for the menu item in the header. We also initialize variables and add the appropriate rows to the **BreadCrumb** repeater.

We will start with the **Journey** page and then complete the remaining pages.

Enabling OnPageLoad actions for the Journey page

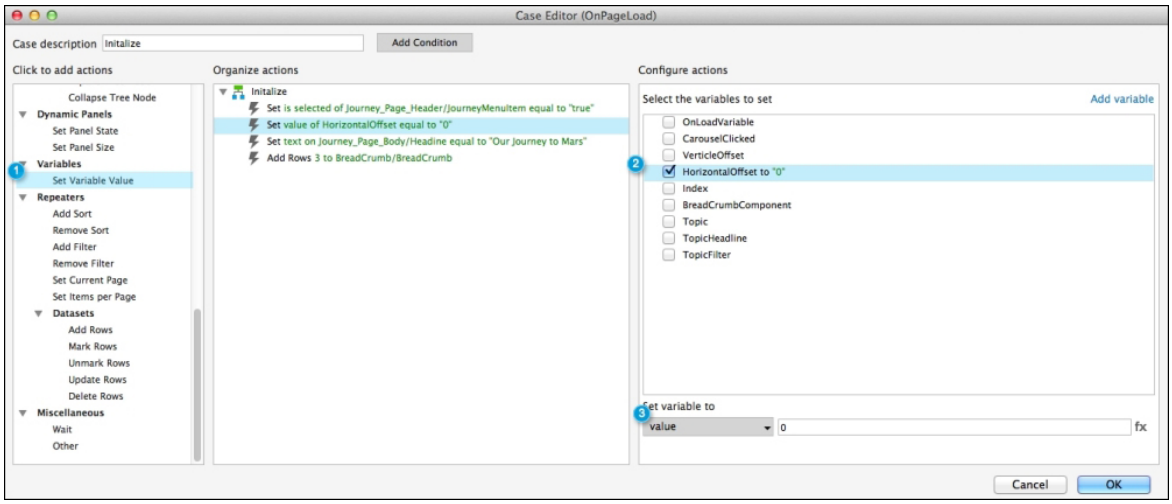
In the **Sitemap** pane, double-click on the Page icon next to the **Journey** page to open in the design area. To enable actions for the **OnPageLoad** event, go in the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `Initialize`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. To set selected to **True** for the **JourneyMenuItem**, perform the following steps:



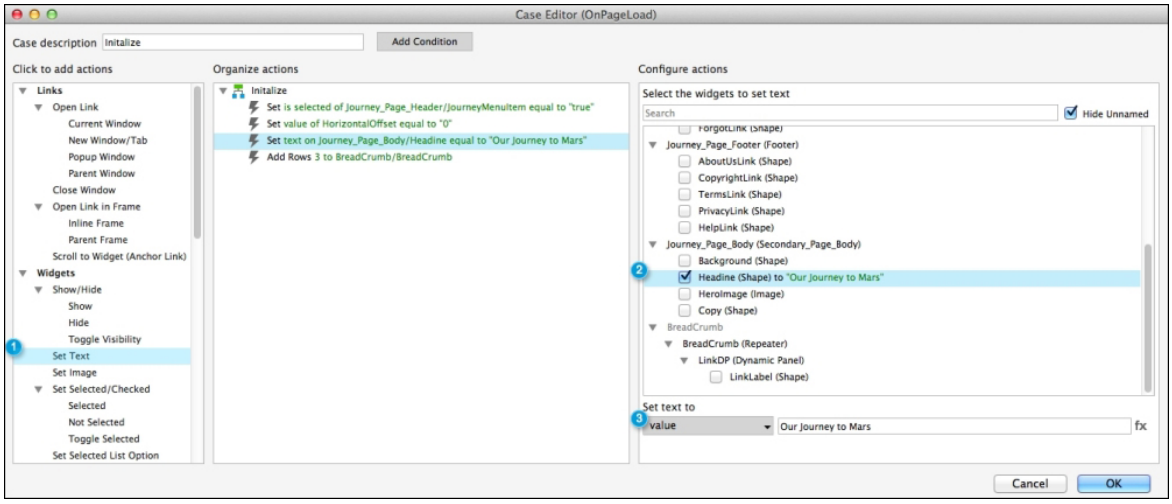
   1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
   2. Under **Configure actions**, click on the checkbox next to **JourneyMenuItem**.
   3. Under **Set selected state to**, verify that the first dropdown is set to **value** and the second dropdown is set to **true**.

2. Next, create the second action. To set the value of the **HorizontalOffset** variable, perform the following steps:



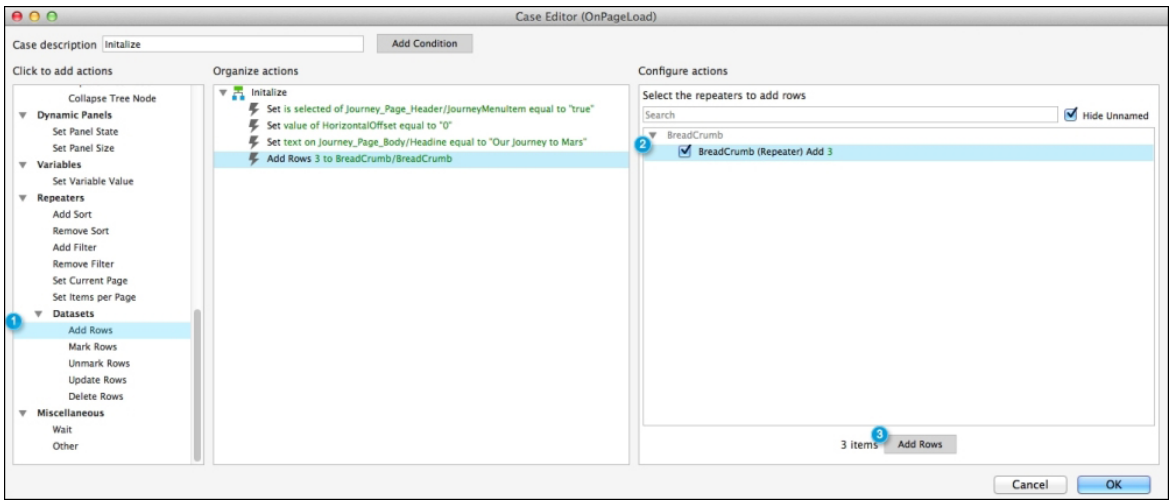
1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
2. Under **Configure actions**, click on the checkbox next to **HorizontalOffset**.
3. Under **Set variable to**, click on the first dropdown and select **value**. In the text field, enter `0`.

3. Now, create the third action. To set the text on **Headline**, perform the following steps:

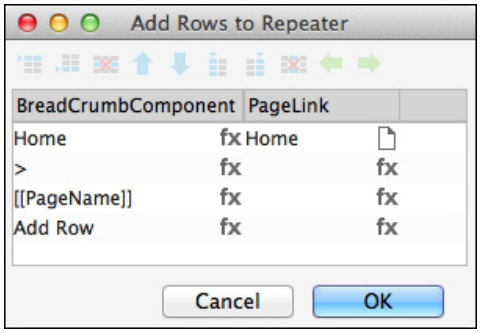


1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
2. Under **Configure actions**, click on the checkbox next to **Headline**.
3. Under **Set text to**, click the first dropdown and click **value**. In the text field, enter `Our Journey to Mars`.

4. Create the fourth action. To add rows to the **BreadCrumb** repeater, perform the following steps:



1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click **Add Rows**.
2. Under **Configure actions**, click on the checkbox next to **BreadCrumb**.
3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update the row values for **BreadCrumbComponent** and add a reference page to **PageLink** using the following screenshot:



> ## Note
>
> To add a reference page for **PageLink**, right-click under **PageLink** in the first row and click on **Reference Page…**. A **Reference Page** dialog box will open. Click on **Home**.

4. Click on **OK**.
5. In **Case Editor**, click on **OK**.

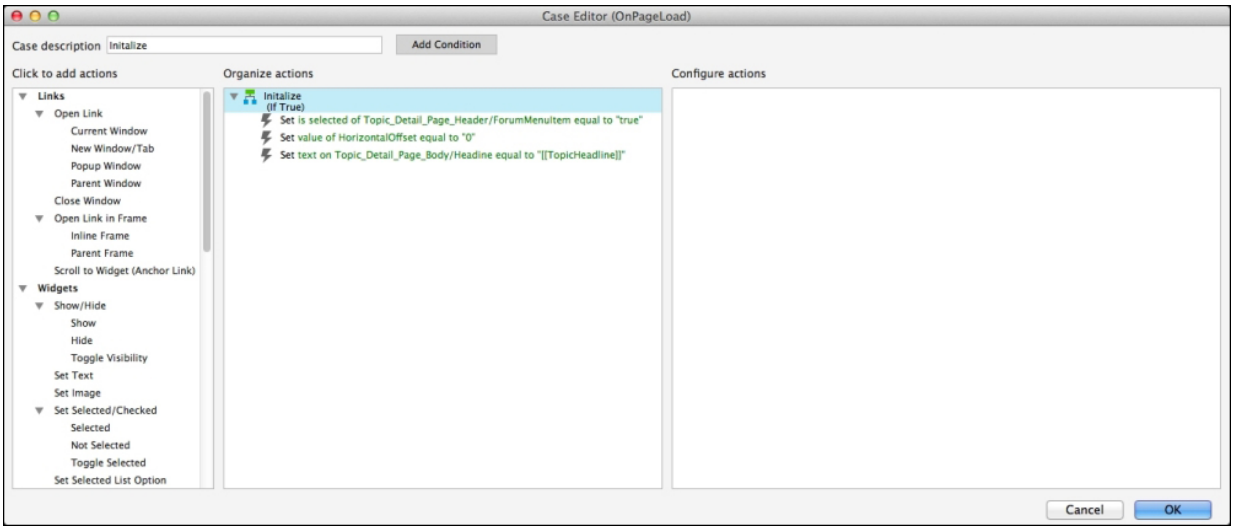
We have completed the design and interactions for the **Journey** page. Now, we will complete the **Topic Detail** page.

Activating OnPageLoad actions for the Topic Detail page

For the **Topic Detail** page, the **OnPageLoad** event has three cases: **Initialize**, **TopicFilterSet**, and **TopicFilterNotSet**.

Let's first see the **Initialize** case:

Once completed, the **Case Editor** for the **Initialize** case will look as follows:



We will now create the **Initialize** case.

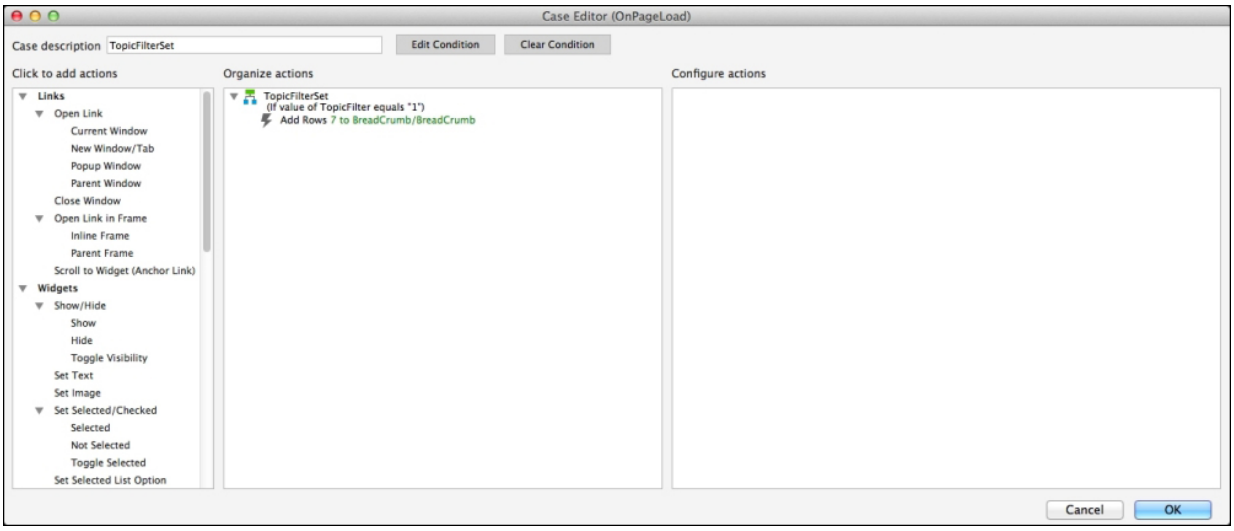
In the **Sitemap** pane, double-click on the Page icon next to the **Topic Detail** page to open in the design area. To enable actions for the **OnPageLoad** event, go in the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `Initialize`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. To set the selected to **True** for the **ForumMenuItem**, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **ForumMenuItem**.
    3. Under **Set selected state to**, verify that the first dropdown is set to **value** and the second dropdown is set to **true**.
2. Now, create the second action. To set the value of the **HorizontalOffset** variable, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **HorizontalOffset**.
    3. Under **Set variable to**, click on the first dropdown and select **value**. In the text field, enter `0`.
3. Create the third action. To set the text on **Headline**, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **Headline**.
    3. Under **Set text to**, click on the first dropdown and select **value**. In the text field, enter `[[TopicHeadline]]`.

4. Click on **OK**.

With the **Initialize** case completed, next we will define the **TopicFilterSet** case.

Once completed, the **Case Editor** for the **TopicFilterCase** case will look as follows:



We will now create the **TopicFilterSet** case.

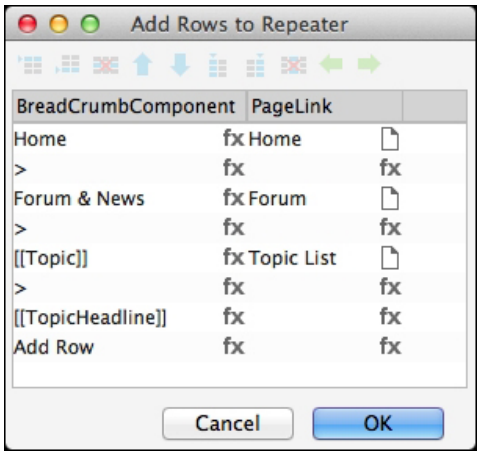
With the **Topic Detail** page open in the design area, go in the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `TopicFilterCase`. In the **Case Editor** dialog, perform the following steps:

1. To add the condition, click on the **Add Condition** button.
2. In the **Condition Builder** dialog box, perform the following steps in the outlined condition box:

    1. In the first dropdown, select **value of variable**.
    2. In the second dropdown, select **TopicFilter**.
    3. In the third dropdown, select **equals**.
    4. In the fourth dropdown, select **value**.
    5. In the text field, enter `1`.
    6. Click on **OK**.

3. Now, create the first action. To add rows to the **BreadCrumb** repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click on **Add Rows**.
    2. Under **Configure actions**, click on the checkbox next to **BreadCrumb**.
    3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update the row values for **BreadCrumbComponent** and add the reference page to **PageLink** using the following screenshot:



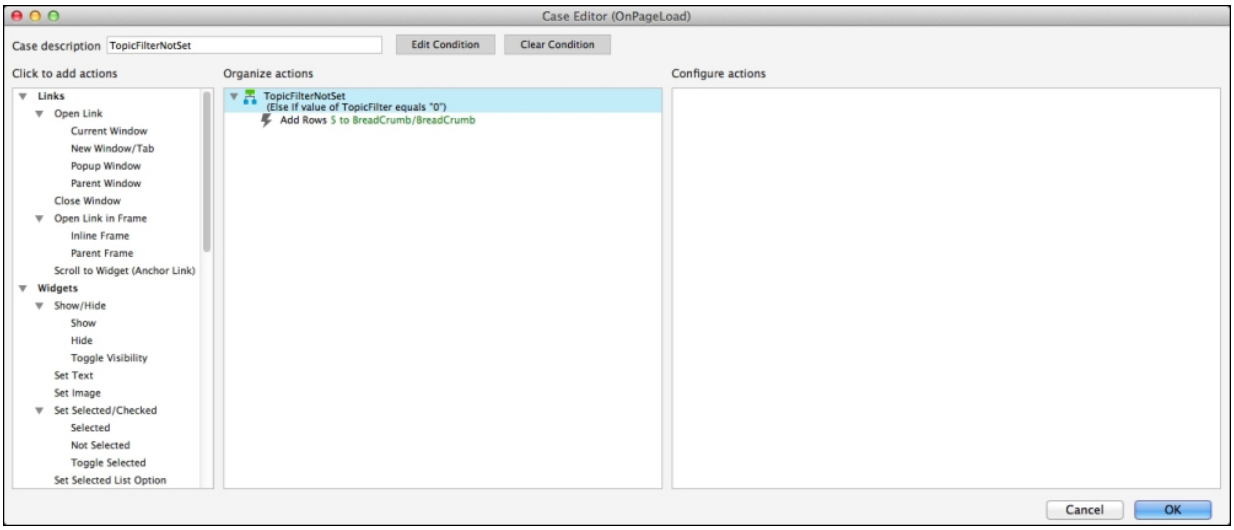
> **Note**
>
> To add a reference page for **PageLink**, right-click under **PageLink** in the first row and click on **Reference Page…**. A **Reference Page** dialog box will open. Click on the corresponding page.

4. Click on **OK**.
5. In the **Case Editor** dialog box, click on **OK**.

4. Under the **OnPageLoad** event, right-click on the **TopicFilterSet** case and click **Toggle IF/ELSE IF**.

With the **TopicFilterSet** case completed, next we will define the **TopicFilterNotSet** case.

Once completed, **Case Editor** for the **TopicFilterNotCase** case will look as follows:



We will now create the **TopicFilterNotSet** case.

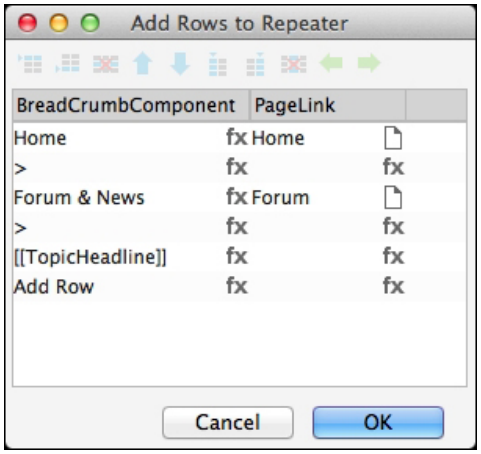
With the **Topic Detail** page open in the design area, go to the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `TopicFilterNotCase`. In the **Case Editor** dialog, perform these steps:

1. To add the condition, click the **Add Condition** button.
2. In the **Condition Builder** dialog box, perform the following steps in the outlined condition box:

    1. In the first dropdown, select **value of variable**.
    2. In the second dropdown, select **TopicFilter**.
    3. In the third dropdown, select **equals**.
    4. In the fourth dropdown, select **value**.
    5. In the text field, enter `0`.
    6. Click on **OK**.

3. Create the first action. To add rows to the **BreadCrumb** repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click on **Add Rows**.
    2. Under **Configure actions**, click on the checkbox next to **BreadCrumb**.
    3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update the row values for **BreadCrumbComponent** and add the reference page to **PageLink** using the following screenshot:



Note

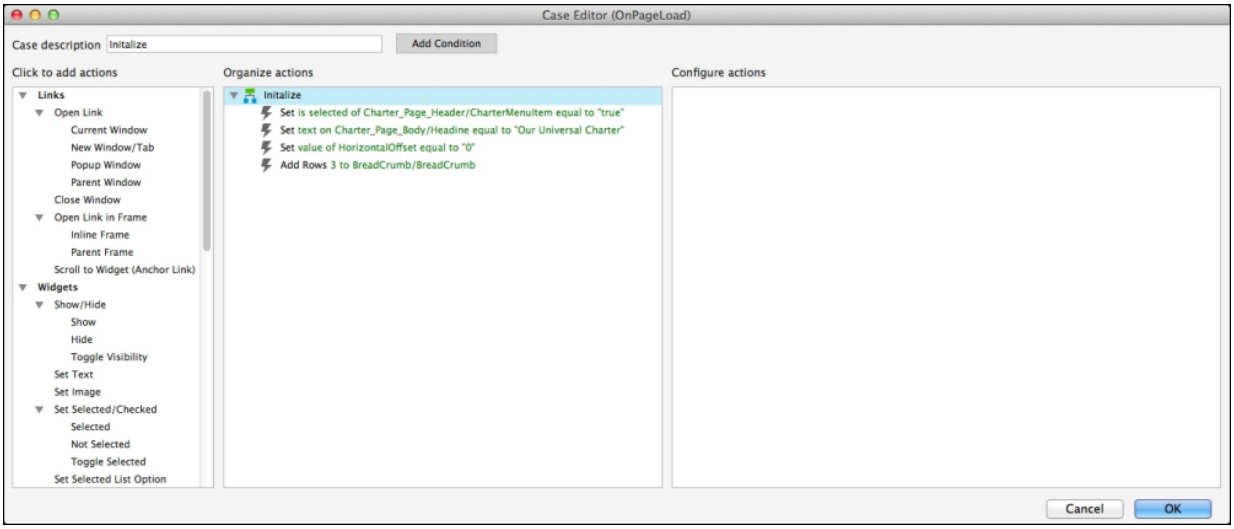
To add a reference page for **PageLink**, right-click under **PageLink** in the first row and click on **Reference Page…**. A **Reference Page** dialog box will open. Click on the corresponding page.

4. Click on **OK**.
5. In **Case Editor**, click on **OK**.

We have completed the design and interactions for the **Topic Detail** page. Now, we will complete the **Charter** page.

Facilitating OnPageLoad actions for the Charter page

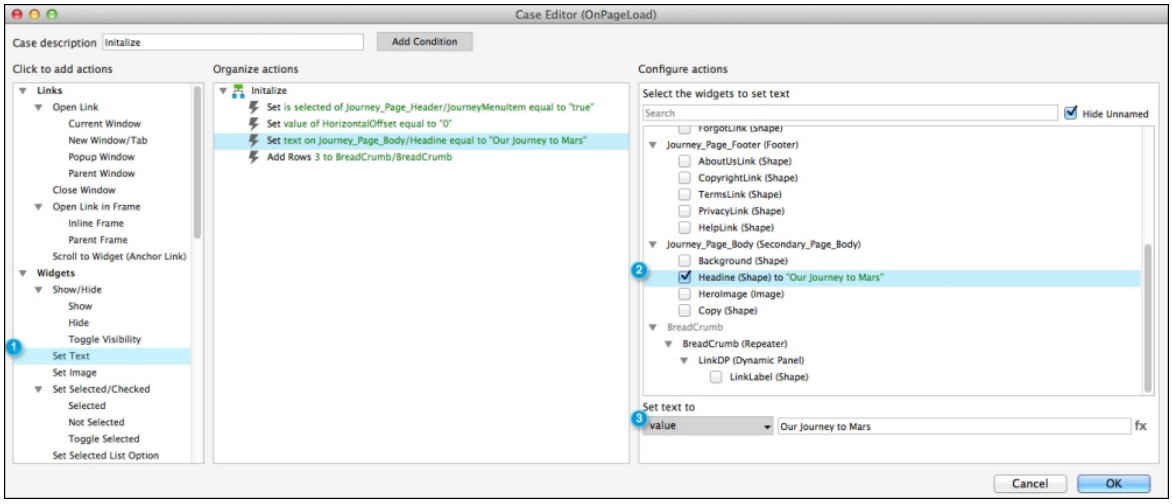
Once completed, the **Case Editor** window for the **Initialize** case will look as follows:



We will now create the **Initialize** case.

In the **Sitemap** pane, double-click on the Page icon next to the **Charter** page to open it in the design area. To enable actions for the **OnPageLoad** event, in the **Page Properties** pane under the design area, click on the **Page Interactions** tab. Click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `Initialize`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. To set the selected to **True** for the **CharterMenuItem**, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **CharterMenuItem**.
    3. Under **Set selected state to**, verify the first dropdown is set to **value** and the second dropdown is set to **true**.
2. Create the second action. To set the text on **Headline**, perform the following steps:



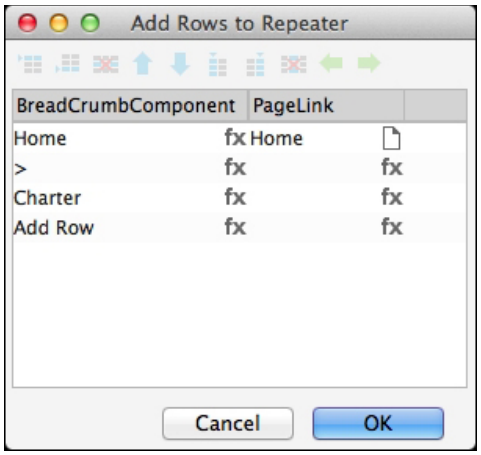
1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
2. Under **Configure actions**, click on the checkbox next to **Headline**.
3. Under **Set text to**, click on the first dropdown and select **value**. In the text field, enter `Our Universal Charter`.

3. Create the third action. To set the value of the **HorizontalOffset** variable, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **HorizontalOffset**.
    3. Under **Set variable to**, click on the first dropdown and select **value**. In the text field, enter `0`.

4. Create the fourth action. To add rows to the **BreadCrumb** repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click on **Add Rows**.
    2. Under **Configure actions**, click on the checkbox next to **BreadCrumb**.
    3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update the row values for **BreadCrumbComponent** and add a reference page to **PageLink** using the following screenshot:



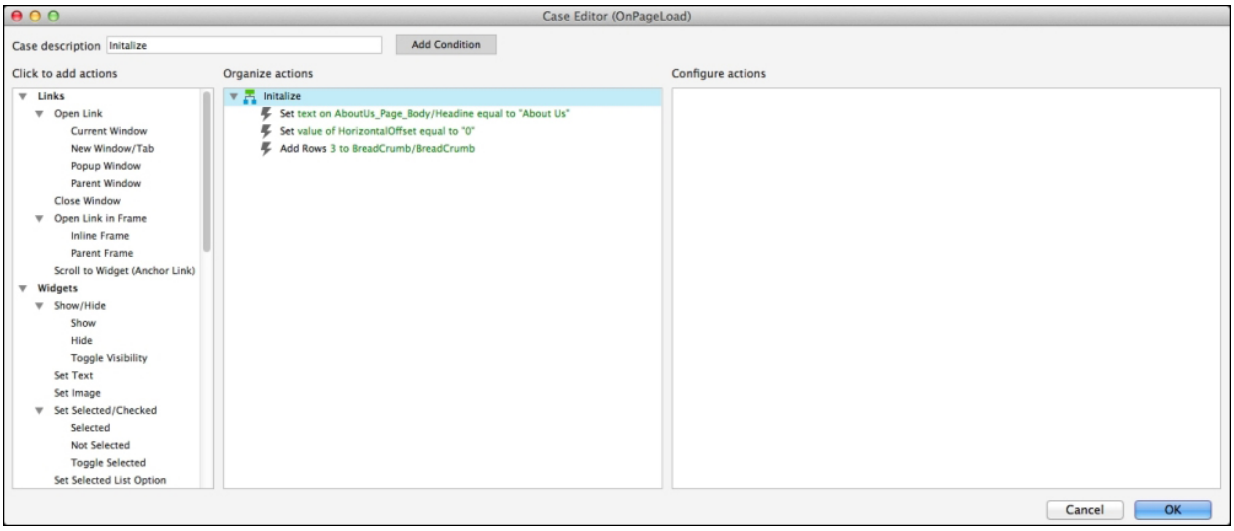
> **Note**
>
> To add a reference page for **PageLink**, right-click under **PageLink** in the first row and click on **Reference Page…**. A **Reference Page** dialog box will open. Click on **Home**.

4. Click on **OK**.
5. In the **Case Editor** window, click on **OK**.

We have completed the design and interactions for the **Charter** page. Now, we will complete the **About Us** page.

Constructing OnPageLoad actions for the About Us page

Once completed, the **Case Editor** dialog box for the **Initialize** case will look as follows:



We will now create the **Initialize** case.

In the **Sitemap** pane, double-click on the Page icon next to the **About Us** page to open in the design area. To enable actions for the **OnPageLoad** event, go in the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `Initialize`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. To set the text on **Headline**, perform the following steps:

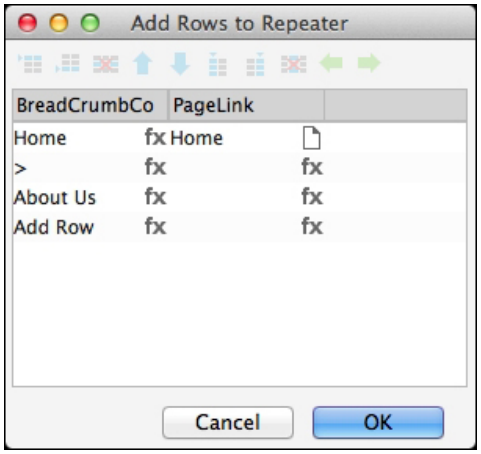
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **Headline**.
    3. Under **Set text to**, click on the first dropdown and click on **value**. In the text field, enter `About Us`.

2. Create the second action. To set value of the **HorizontalOffset** variable, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **HorizontalOffset**.
    3. Under **Set variable to**, click the first dropdown and click on **value**. In the text field, enter `0`.

3. Create the third action. To add rows to the **BreadCrumb** repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click on **Add Rows**.
    2. Under **Configure actions**, click on the checkbox next to **BreadCrumb**.
    3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update the row values for **BreadCrumbComponent** and add a reference page to **PageLink** using the following screenshot:



> **Note**
>
> To add a reference page for **PageLink**, right-click under **PageLink** in the first row and click on **Reference Page…**. A **Reference Page** dialog box will open. Click on **Home**.

4. Click on **OK**.
5. In **Case Editor**, click on **OK**.

We have completed the design and interactions for the **About Us** page. Now, we will complete the design and interactions for the **Forum** page.

Manufacturing design and interactions for the Forum page

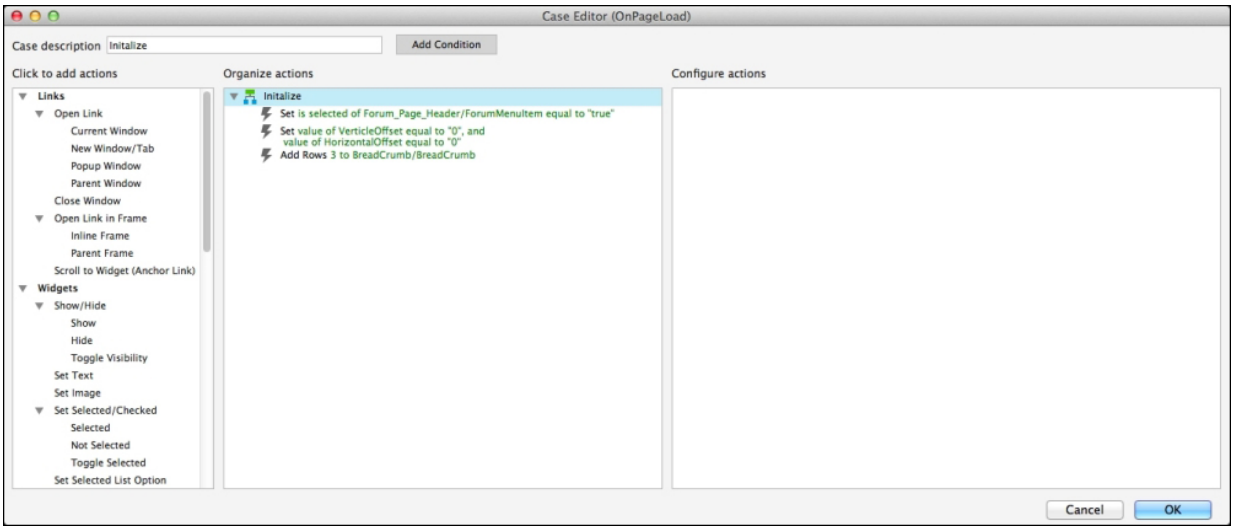
The **Forum** page uses the **Header**, **Forum**, and **Footer** masters. To complete the design, perform the following steps:

1. In the **Masters** pane, double-click on the Page icon next to the **Forum** page to open in the design area.
2. In the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. In the **Widgets** pane, drag the **Rectangle** widget and place it at the coordinates (10,140). In the toolbar, change **w:** to `940` and **h:** to `370`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Background`.
4. In the **Widgets** pane, drag the **Heading 1** widget and place it at the coordinates (20,147). Type `Forum & News`. In the toolbar, change **w:** to `225` and **h:** to `37`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Headline`.
5. In the **Widgets** pane, drag the **Image** widget and place at the coordinates (20,190). In the toolbar, change **w:** to `920` and **h:** to `230`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `HeroImage`.
6. In the **Widgets** pane, drag the **Paragraph** widget and place at the coordinates (20,430). In the toolbar, change **w:** to `920` and **h:** to `73`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Copy`. Adjust the **Paragraph** text to fit within the **Section1_Background** rectangle.
7. In the **Masters** pane, drag the **Forum** master and place it at the coordinates (20,530). In the **Widget Interactions and Notes** pane, click on the **Repeater Name** field and type `ForumRepeater`.
8. In the **Masters** pane, drag the **Footer** master and drop it anywhere in the design area.

We have completed the design for the **Forum** page. Now we will complete the interactions.

Defining OnPageLoad actions for the Forum page

Once completed, the **Case Editor** window for the **Initialize** case will look as follows:



We will now create the **Initialize** case.

In the **Sitemap** pane, double-click on the Page icon next to the **Forum** page to open it in the design area. To enable actions for the **OnPageLoad** event, go in the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `Initialize`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. To set selected to **True** for the **ForumMenuItem**, perform the following steps:

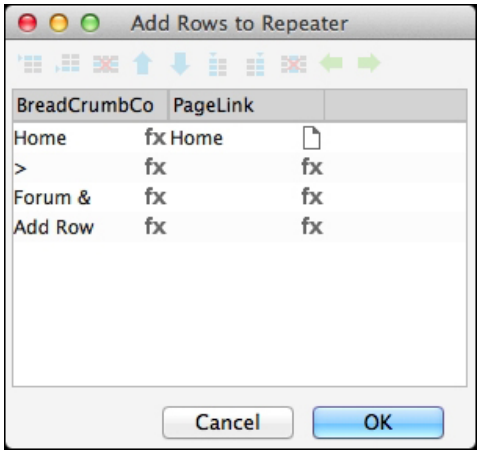
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click the checkbox next to **ForumMenuItem**.
    3. Under **Set selected state to**, verify that the first dropdown is set to `value` and the second dropdown is set to `true`.

2. Create the second action. To set value of the **VerticleOffset** and **HorizontalOffset** variables, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **VerticleOffset**.
    3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `0`.
    4. Under **Configure actions**, click on the checkbox next to **HorizontalOffset**.
    5. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `0`.

3. Create the third action. To add rows to the **BreadCrumb** repeater, perform the following steps:

1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click **Add Rows**.
2. Under **Configure actions**, click on the checkbox next to **BreadCrumb**.
3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update the row values for **BreadCrumbComponent** and add a reference page to **PageLink** using the following screenshot:



> ## Note
>
> To add a reference page for **PageLink**, right-click under **PageLink** in the first row and click on **Reference Page…**. A **Reference Page** dialog box will open. Click on **Home**.

4. Click on **OK**.
5. In the **Case Editor** window, click on **OK**.

We have completed the design and interactions for the **Forum** page. Now, we will complete the design and interactions for the **Topic List** page.

Fabricating design and interactions for the Topic List page

The **Topic List** page uses the **Header**, **Forum**, and **Footer** masters. To complete the design, perform the following steps:

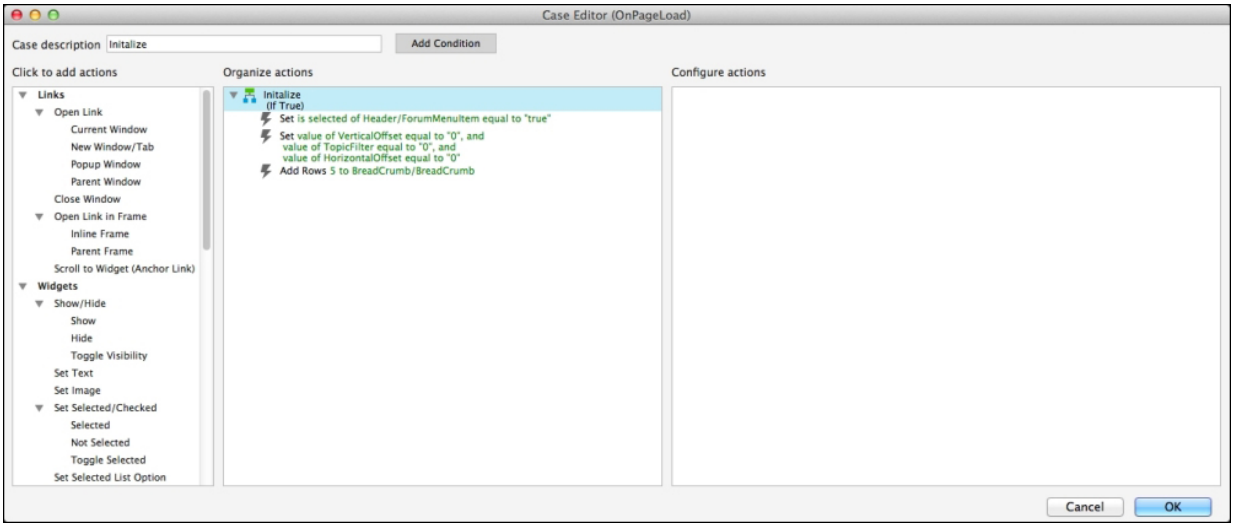
1. In the **Masters** pane, double-click on the Page icon next to the **Topic List** page to open in the design area.
2. In the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. Again in the **Masters** pane, drag the **Forum** master and place at the coordinates (20,140). In the **Widget Interactions and Notes** pane, click on the **Repeater Name** field and type `TopicListRepeater`.
4. In the **Masters** pane, drag the **Footer** master and drop it anywhere in the design area.

We have completed the design for the **Topic List** page. Now, we will complete the interactions.

Defining OnPageLoad actions for the Topic List page

The **Topic List** page has two cases for the **OnPageLoad** event: **Initialize** and **SetAllTopicFilter**. We will first create the **Initialize** case.

Once completed, the **Case Editor** window for the **Initialize** case will look as follows:



We will now create the **Initialize** case.

Creating the Initialize case for the OnPageLoad event

In the **Sitemap** pane, double-click on the Page icon next to the **Topic List** page to open in the design area. To enable actions for the **OnPageLoad** event, go in the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `Initialize`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. To set the selected to **True** for the **ForumMenuItem**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** dropdown menu, click on **Selected**.
    2. Under **Configure actions**, click the checkbox next to **ForumMenuItem**.
    3. Under **Set selected state to**, verify that the first dropdown is set to `value` and the second drop down is set to `true`.

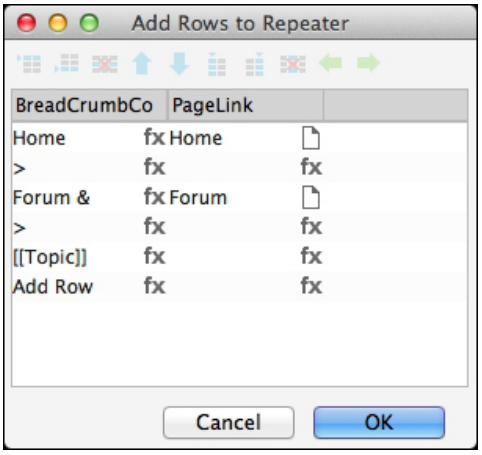
2. Create the second action. To set the value of the **VerticleOffset**, **TopicFilter**, and **HorizontalOffset** variables, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.

2. Under **Configure actions**, click on the checkbox next to **VerticleOffset**.
3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `0`.
4. Under **Configure actions**, click on the checkbox next to **TopicFilter**.
5. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `0`.
6. Under **Configure actions**, click on the checkbox next to **HorizontalOffset**.
7. Under **Set variable to**, click the first dropdown and click **value**. In the text field enter `0`.

3. Create the third action. To add rows to the **BreadCrumb** repeater, perform the following steps:

1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click on **Add Rows**.
2. Under **Configure actions**, click on the checkbox next to **BreadCrumb**.
3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update the row values for **BreadCrumbComponent** and add a reference page to **PageLink** using the following screenshot:



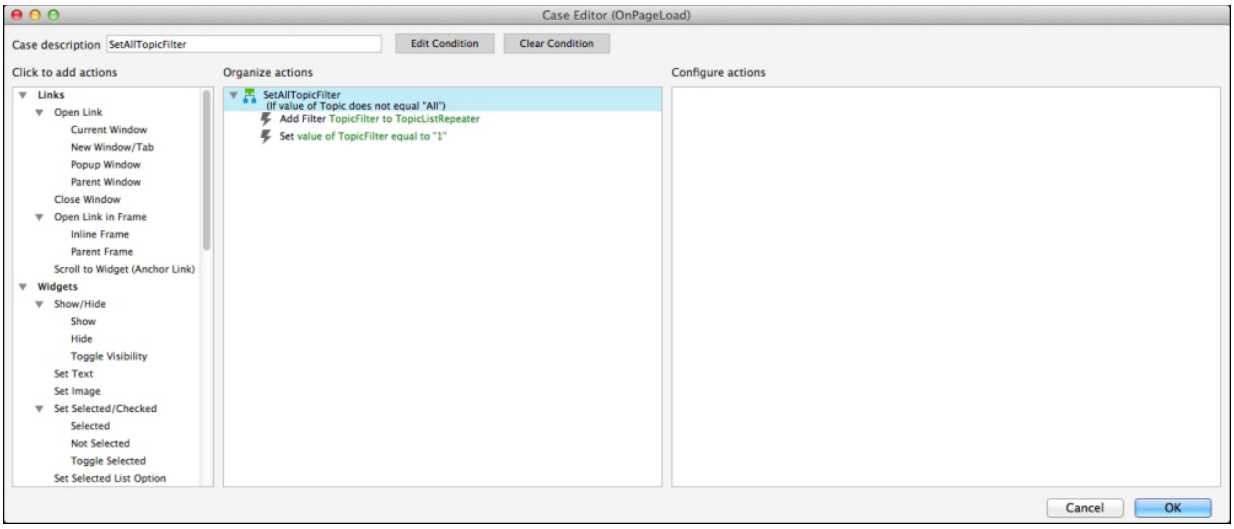
> ## Note
>
> To add a reference page for **PageLink**, right-click under **PageLink** in the first row and click on **Reference Page…**. A **Reference Page** dialog box will open. Click on **Home**.

4. Click on **OK**.
5. In the **Case Editor** window, click on **OK**.

Next, we will create the **SetAllTopicFilter** case.

Adding the SetAllTopicFilter case for the OnPageLoad event

Once completed, the **Case Editor** window for the **SetAllTopicFilter** case will look as follows:



With the **Topic Detail** page open in the design area, go in the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `TopicFilterNotCase`. In the **Case Editor** dialog, perform the following steps:

1. To add the condition, click on the **Add Condition** button.
2. In the **Condition Builder** dialog box, perform the following steps in the outlined condition box:

    1. In the first dropdown, select **value of variable**.
    2. In the second dropdown, select **Topic**.
    3. In the third dropdown, select **equals**.
    4. In the fourth dropdown, select **value**.
    5. In the text field, enter `All`.
    6. Click on **OK**.

3. Create the first action. To add filer to the **TopicList** repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Repeaters** drop-down menu and click on **Add Filter**.
    2. Under **Configure actions**, click on the checkbox next to **TopicListRepeater**. In the **Name** field, type `TopicFilter`. In the **Rule** field, type `[[Item.TopicCategory == Topic]]`.

4. Create the second action. To set value of the **TopicFilter** variable, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **TopicFilter**.
    3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `1`.

4. Click on **OK**.
5. Under the **OnPageLoad** event, right-click on the **TopicFilterSet** case and click on **Toggle IF/ELSE IF**.

We have completed the design and interactions for the **Topic List** page. Now, we will complete the design and interactions for the **Terms**, **Privacy**, and **Help** pages.

Completing the design and interactions for the Terms, Privacy, and Help pages

The **Terms**, **Privacy**, and **Help** pages use the **Header**, **Informational_Page_Body**, and **Footer** masters. To complete the design, perform the following steps:

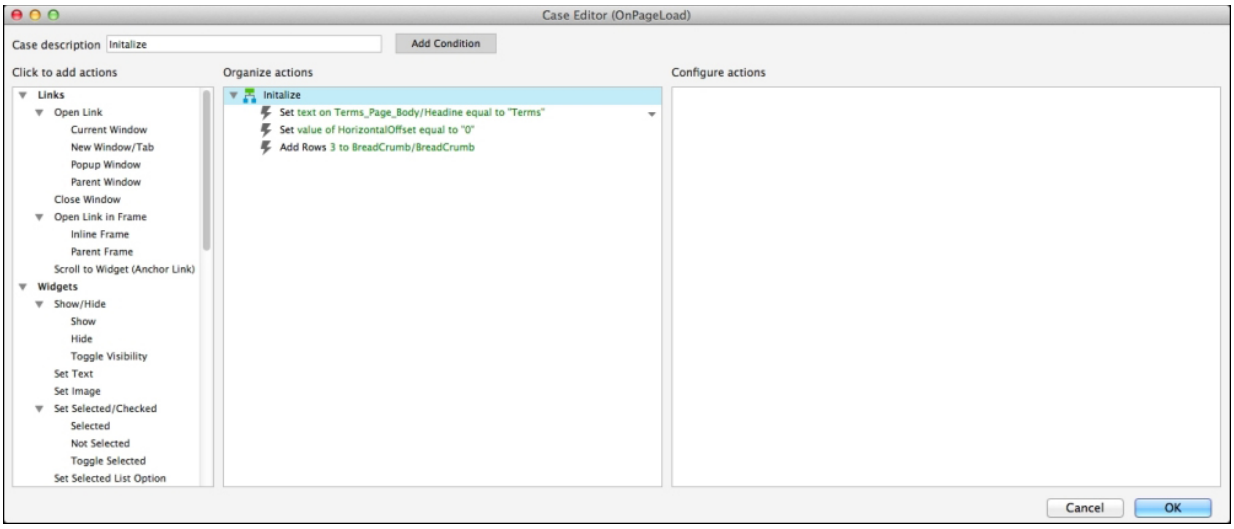
1. In the **Masters** pane, double-click on the Page icon next to the **Terms** page to open it in the design area.
2. In the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. In the **Masters** pane, drag the **Informational_Page_Body** master and drop it anywhere in the design area.
4. In the **Masters** pane, drag the **Footer** master and drop it anywhere in the design area.
5. Repeat steps 1-4 to complete the designs for the **Privacy** and **Help** pages. As mentioned in step 1, open the **Privacy** and **Help** pages one by one and follow the next steps until step 4.

Page to Open **(Step 1)**
Privacy
Help

We have completed the design for the **Terms**, **Privacy**, and **Help** pages. Now, we will complete the **OnPageLoad** interactions for each of the pages.

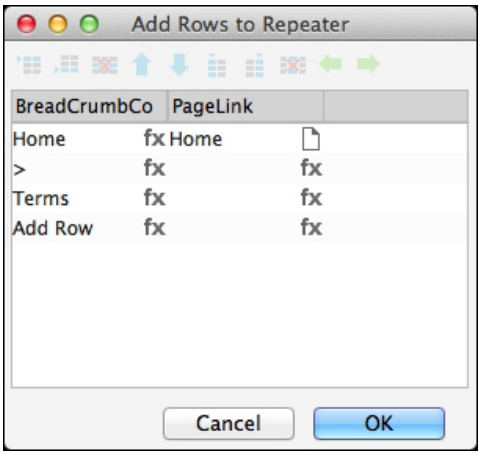
Creating the Initialize case for the OnPageLoad event for the Terms page

Once completed, the **Case Editor** window for the **Initialize** case will look as follows:



In the **Sitemap** pane, double-click on the Page icon 🗋 next to the **Terms** page to open it in the design area. To enable actions for the **OnPageLoad** event, go in the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `Initialize`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. To set the text on **Headline**, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **Headline**.
    3. Under **Set text to**, click on the first dropdown and click on **value**. In the text field, enter `Terms`.
2. Create the second action. To set value of the **HorizontalOffset** variable, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **HorizontalOffset**.
    3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `0`.
3. Create the third action. To add rows to the **BreadCrumb** repeater, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click on **Add Rows**.
    2. Under **Configure actions**, click on the checkbox next to **BreadCrumb**.
    3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update the row values for **BreadCrumbComponent** and add a reference page to **PageLink** using the following screenshot:



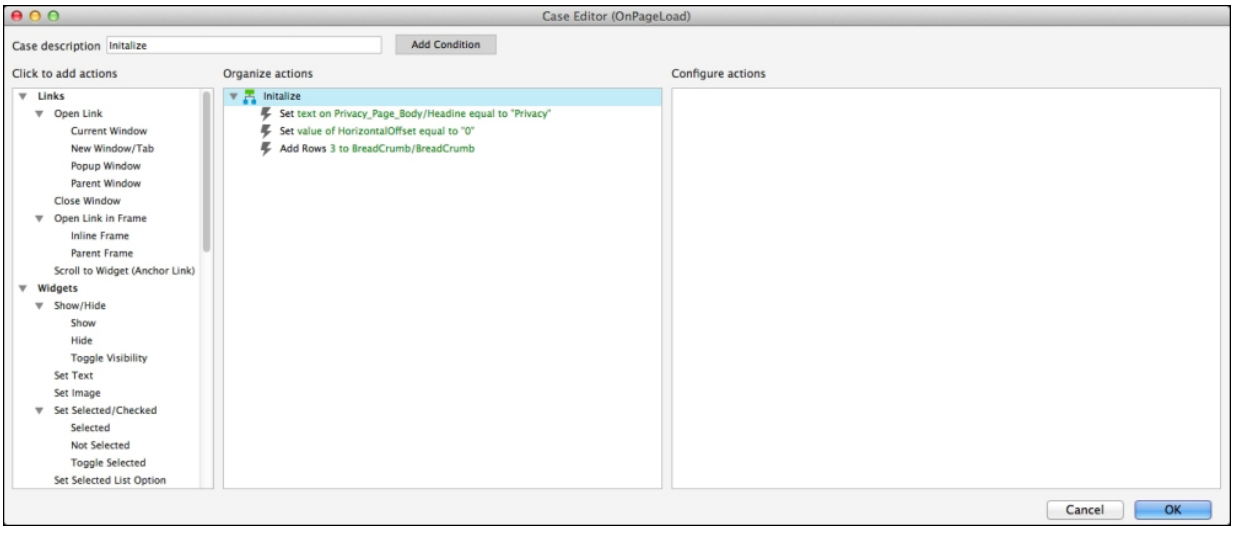
> **Note**
>
> To add a reference page for **PageLink**, right-click under **PageLink** in the first row and click on **Reference Page…**. A **Reference Page** dialog box will open. Click on **Home**.

4. Click on **OK**.
5. In the **Case Editor** window, click on **OK**.

We have completed the interactions for the **Terms** page. Now, we will complete the **OnPageLoad** interactions for the **Privacy** page.

Creating the Initialize case for the OnPageLoad event for the Privacy page

Once completed, the **Case Editor** window for the **Initialize** case will look as follows:



In the **Sitemap** pane, double-click on the Page icon next to the **Privacy** page to open it in the design area. To enable actions for the **OnPageLoad** event, go in the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on

Add Case…. A **Case Editor** dialog box will open. In the **Case description** field, type `Initialize`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. To set the text in **Headline**, perform the following steps:

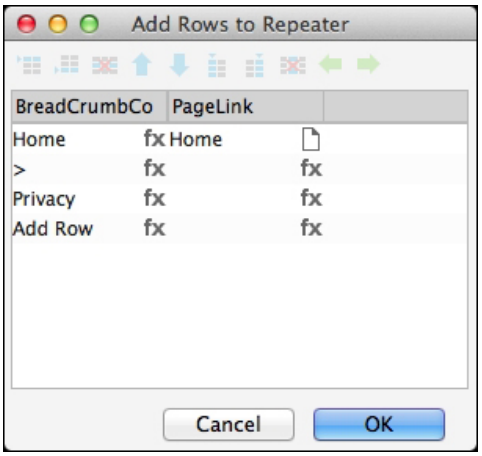
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **Headline**.
    3. Under **Set text to**, click on the first dropdown and click on **value**. In the text field, enter `Privacy`.

2. Create the second action. To set the value of the **HorizontalOffset** variable, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **HorizontalOffset**.
    3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `0`.

3. Create the third action. To add rows to the **BreadCrumb** repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click on **Add Rows**.
    2. Under **Configure actions**, click on the checkbox next to **BreadCrumb**.
    3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update the row values for **BreadCrumbComponent** and add a reference page to **PageLink** using the following screenshot:



> **Note**

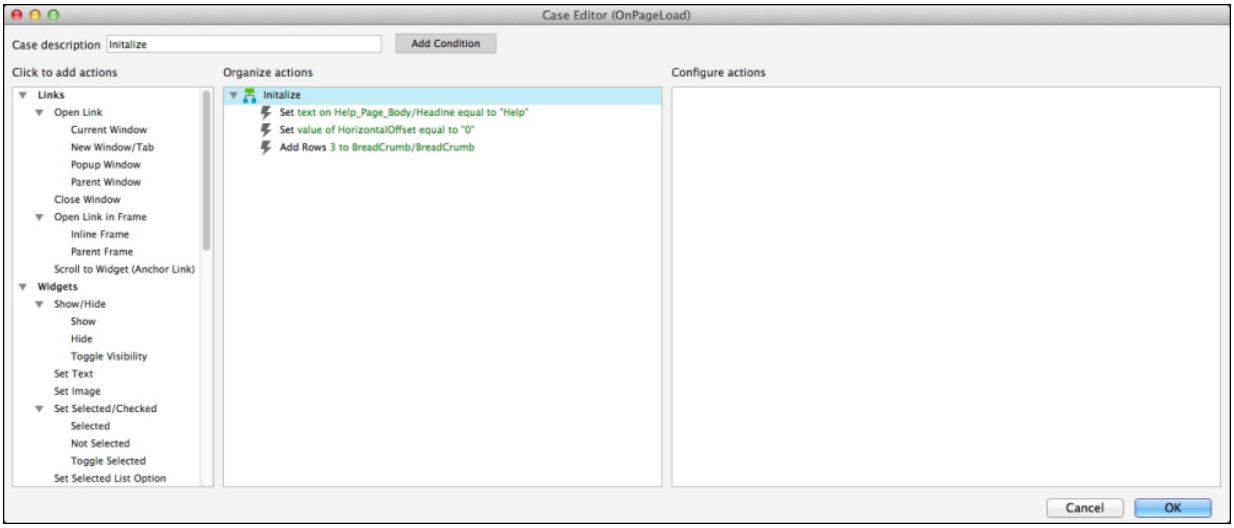
To add a reference page for **PageLink**, right-click under **PageLink** in the first row and click on **Reference Page…**. A **Reference Page** dialog box will open. Click on **Home**.

4. Click on **OK**.
5. In the **Case Editor** window, click on **OK**.

We have completed the interactions for the **Privacy** page. Now, we will complete the **OnPageLoad** interactions for the **Help** page.

Creating the Initialize case for the OnPageLoad event for the Help page

Once completed, the **Case Editor** window for the **Initialize** case will look as follows:



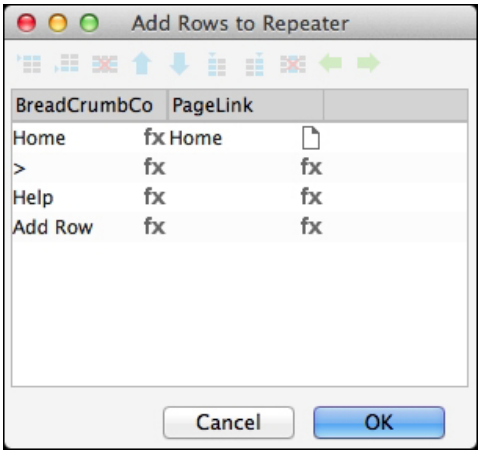
In the **Sitemap** pane, double-click on the Page icon next to the **Help** page to open it in the design area. To enable actions for the **OnPageLoad** event, go in the **Page Properties** pane under the design area and click on the **Page Interactions** tab. Click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case description** field, type `Initialize`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. To set text in headline, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **Headline**.
    3. Under **Set text to**, click on the first dropdown and click on **value**. In the text field, enter `Help`.
2. Create the second action. To set value of the **HorizontalOffset** variable, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **HorizontalOffset**.

3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `0`.

3. Create the third action. To add rows to the **BreadCrumb** repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click on **Add Rows**.
    2. Under **Configure actions**, click on the checkbox next to **BreadCrumb**.
    3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update the row values for **BreadCrumbComponent** and add a reference page to **PageLink** using the following screenshot:



> ## Note
>
> To add a reference page for **PageLink**, right-click under **PageLink** in the first row and click on **Reference Page…**. A **Reference Page** dialog box will open. Click on **Home**.

    4. Click on **OK**.
    5. In the **Case Editor** window, click on **OK**.

Congratulations, we have completed all pages of our community site.

Summary

In this chapter, we learned how to create masters to simplify our site design. We created a **Header** and **Footer** master that was used on every page of our community site. For the **Home** page, we created an interactive carousel, a call to action area, and social media feed. To assist with navigation, we created a dynamic BreadCrumb.

To assist in quickly building our remaining pages, we created three types of page body masters. The first was a secondary page body that contained a headline, hero image, and supportive copy. The second was an informational page body that contained two separate sections made up of a headline and copy. The third was a forum repeater that was leveraged for the **Forum** and **Topic List** pages.

In the next chapter, we will explore the building of a blog with content streams.

Chapter 3. Constructing a Blog

Blogs can range from an individual's online journal to a professionally edited **Multi-Author Blog** (**MAB**) for media outlets. Like our community site, blogs organize content into categories and topics or posts. Privileged users can add additional topics. Registered users join the conversation stream by posting comments on the topics. Typically, conversation streams show the most recent posts first.

As implied, there can be a lot of content to moderate and manage on an interactive blog site. One of the most popular **Content Management Systems** (**CMS**) for blogs is WordPress. According to builtwith.com, over 43% of sites using a CMS have chosen WordPress.

In this chapter, you will learn about:

- Reviewing our project
- Planning our blog
- Updating and building our masters
- Repurposing pages in our sitemap

Reviewing our project

After exploring WordPress.org for a few minutes, we find that WordPress has numerous plugins and themes providing additional flexibility to support our design. Based on our research, we determine that WordPress is a perfect fit for a new blog project.

Background information for our blog

A songwriter who needs a blog contacts our agency. Our songwriter wants something simple and easy to update that enables registered users to comment on posts. She wants the home page to display at

the front and center of her latest posts. She would also like the main navigation to have the following three menu items:

- **Random Musings**
- **Accolades and News**
- **About**

In addition, our client requests to see a prototype highlighting the following user stories:

- **Registration**: As a new user, I want to easily sign up for an account.
- **Create New Posts**: As an administrator, I want to easily create new posts.
- **Comment on Posts**: As a registered user, I want to easily add my thoughts (that is, comments) on existing posts.

Our client wants to see wires and a prototype of the experience as fast as humanly possible, and she is on a tight budget. Time is ticking, let's get started!

Planning our blog

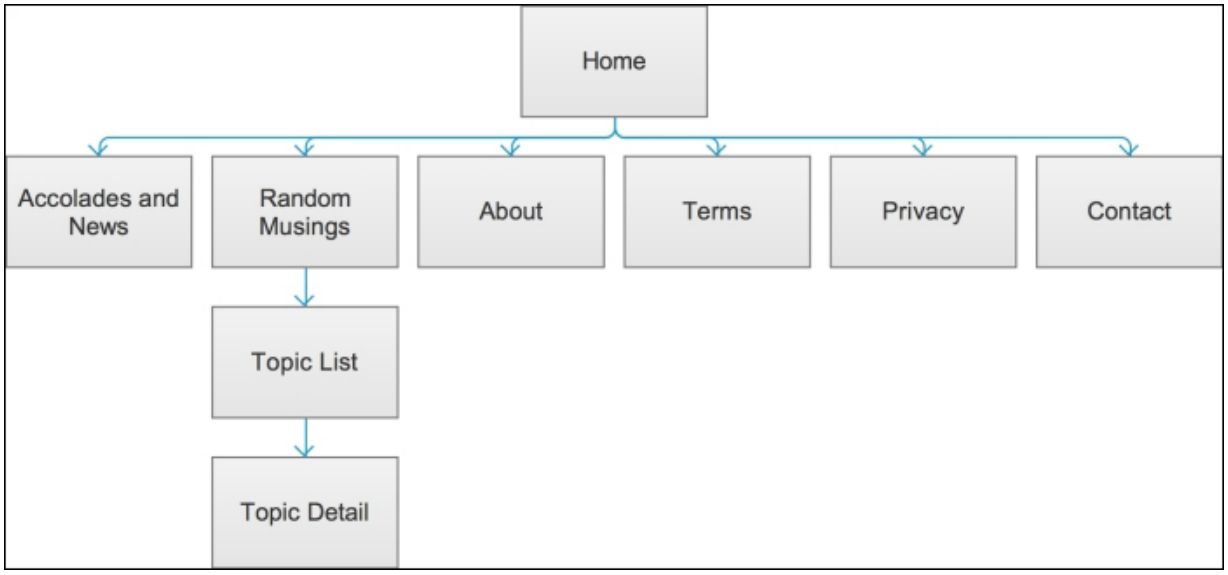
We will start by creating a sitemap and generate flow documents to support the required use cases. After reviewing our sitemap and flow, we can then identify masters that need to be developed and can repurpose.

We are now ready to inspect the sitemap and generate the forum and news flow document.

Inspecting our sitemap

Based on the background information provided by our client, we determine that we can leverage the structure and number of widgets developed for our community site in the previous chapter. First, we copied the Axure RP file created in Chapter 2, *Building a Community Site*, and renamed it in this chapter. Then we update the sitemap by deleting, renaming, and organizing pages in the **Sitemap** pane.

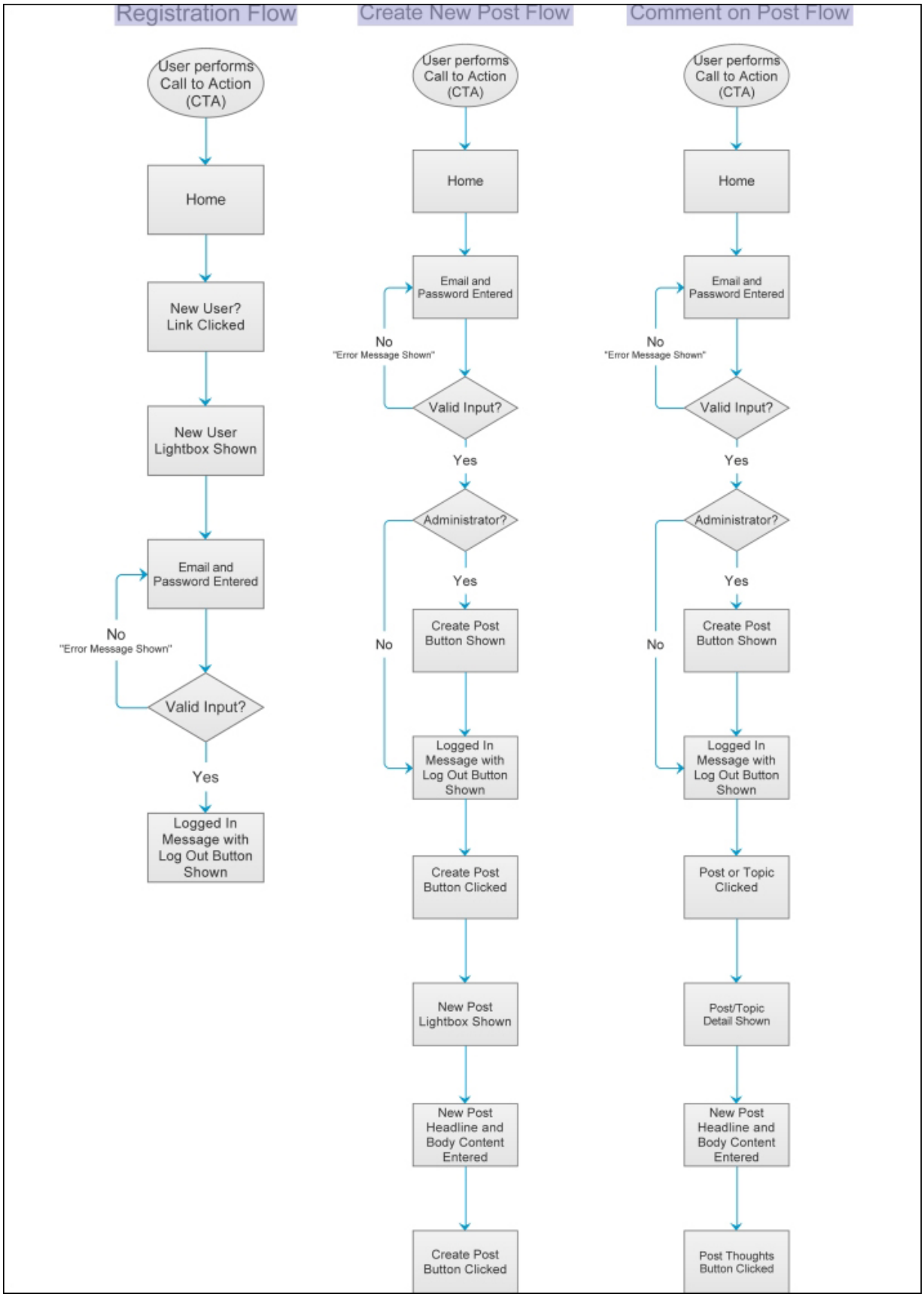
The new sitemap for our blog is as follows:



We will now generate and review the flow document supporting the required use cases.

Reviewing the registration and post flows

Based on the background information provided by our client, we created three flows, **Registration**, **Create New Post**, and **Comment on Post** as follows:


Registration Flow
User performs Call to Action (CTA)
Home
New User? Link Clicked
New User Lightbox Shown
Email and Password Entered
No
"Error Message Shown"
Valid Input?
Yes
Logged In Message with Log Out Button Shown
Create New Post Flow
User performs Call to Action (CTA)
Home
Email and Password Entered
No
"Error Message Shown"
Valid Input?
Yes
Administrator?
Yes
No
Create Post Button Shown
Logged In Message with Log Out Button Shown
Create Post Button Clicked
New Post Lightbox Shown
New Post Headline and Body Content Entered
Create Post Button Clicked
Comment on Post Flow
User performs Call to Action (CTA)
Home
Email and Password Entered
No
"Error Message Shown"
Valid Input?
Yes
Administrator?
Yes
No
Create Post Button Shown
Logged In Message with Log Out Button Shown
Post or Topic Clicked
Post/Topic Detail Shown
New Post Headline and Body Content Entered
Post Thoughts Button Clicked

We are now ready to update and build new masters.

Updating and building our masters

We determine that with modification, we can use all of the masters from the previous community site. To support our new use cases, we need additional registration variables, a master to support user registration, and interactions for the creation of, and to comment on, posts.

Next we will create global variables and add new masters, as well as enhance the design and interactions for each master.

Creating additional global variables

Based on project requirements, we identified that nine global variables will be required. To create global variables, on the main menu click on **Project** and then click on **Global Variables…**. In the **Global Variables** dialog, perform the following steps:

1. Click the green **+** sign and type `Email`. Click on the **Default Value** field and type `songwriter@test.com`.
2. Repeat step 1 eight more times to create additional variables using the following table for the **Variable Name** and **Default Value** fields:

Variable Name	**Default Value**
Password	`Grammy`
UserEmail	
UserPassword	
LoggedIn	`No`
TopicIndex	`0`

Variable Name	Default Value
UserText	
NewPostTopic	
NewPostHeadline	

3. Click on **OK**.

With our global variables created, we are now ready to create new masters, as well as update the design and interactions for existing masters. We will start by adding masters to the **Masters** pane.

Adding masters to the Masters pane

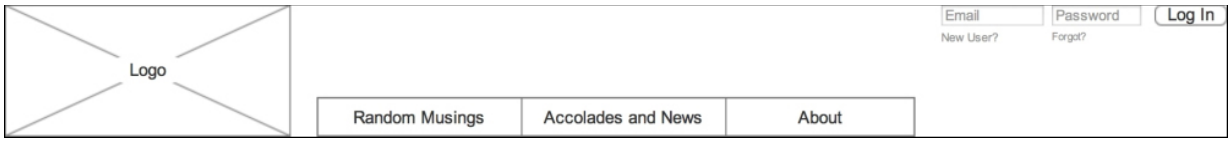
We will add a total of two masters to the **Masters** pane. To create our masters, perform the following steps:

1. In the **Masters**pane, click on the, **Add Master** icon , type `PostCommentary` and press *Enter*.
2. Again, in the **Masters** pane, click on the **Add Master** icon , type `NewPost` and press *Enter*.
3. In the same **Masters** pane, right-click on the icon next to the **Header** master, mouse over **Drop Behavior** and click on **Lock to Master Location**.

We are now ready to remodel the existing masters and complete the design and interactions for our new masters. We will start with the **Header** master.

Enhancing our Header master

Once completed, the **Header** master will look as follows:



To update the **Header** master, we will add an **ErrorMessage** label, delete the **Search** widgets, and update the menu items. To update widgets on the **Header** master, perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Header** master to open in the design area.
2. In the **Widgets** pane, drag the **Label** widget and place it at coordinates (730,0).
3. Now, select the **Text Field** widget and type `Your email or password is incorrect.`.
4. In the **Widget Interactions and Notes** pane, click in the **Shape Name** field and type `ErrorMessage`.
5. In the **Widget Properties and Style** pane, with the **Style** tab selected, scroll to **Font** and perform the following steps:

    1. Change the font size to `8`.
    2. Click on the down arrow next to the Text Color icon. In the drop-down menu, in the **#** text field, enter `FF0000`.
    3. In the toolbar, click on the checkbox next to **Hidden**.

6. Click on the **EmailTextField** at coordinates (730,10). If text is displayed on the text field, right-click and click **Edit Text**. All text on the widget will be highlighted, click on **Delete**. In the **Widget Properties and Style** pane, with the **Properties** tab selected, scroll to **Text Field** and perform the following steps:

    1. Next to **Hint Text**, enter `Email`.
    2. Click **Hint Style**. In the **Set Interaction Styles** dialog box, click on the checkbox next to **Font Color**. Click on the down arrow next to the Text Color icon. In the drop-down menu, in the **#** text field, enter `999999`. Click on **OK**.

7. Click on the **PasswordTextField** at coordinates (815,10). If text is displayed on the text field, right-click and click on **Edit Text**. All text on the widget will be highlighted, press *Delete*.
8. In the **Widget Properties and Style** pane, with the **Properties** tab selected, scroll to **Text Field** and perform the following steps:

    1. Click on the drop-down menu next to **Type** and select `Password`.
    2. Next to **Hint Text**, enter `Password`.
    3. Click on **Hint Style**. In the **Set Interaction Styles** dialog box, click on the checkbox next to **Font Color**. Click on the down arrow next to the Text Color icon . In the drop-down menu, in the **#** text field, enter `999999`.
    4. Click on **OK**.
9. Click on the **SearchTextField** at coordinates (730,82) and then on **Delete**.
10. Click on the **SearchButton** at coordinates (890,80) and then on **Delete**.

Next, we will convert all the Log In widgets into a dynamic panel named **LoginDP**. The **LoginDP** will allow us to transition between states and show different content when a user logs in. To create the **LoginDP**, in our header, select the following widgets:

Named Widget	**Coordinates**
ErrorMessage	(730,0)
EmailTextField	(730,10)
PasswordTextField	(815,10)
LogInButton	(894,10)
NewUserLink	(730,30)
ForgotLink	(815,30)

With the preceding six widgets selected, right-click and click **Convert to Dynamic Panel**.

In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `LogInDP`. All the Log In widgets are now on **State1** of the **LogInDP**.

We will now add widgets to **State2** for the **LogInDP**. With the Log In widgets converted into the **LogInDP**, we will now add and design **State2**. In the **Widget Manager** pane, under the **LogInDP**, right-click on **State1,** and in the menu, click on **Add State**. Click on the State icon beside **State2** twice, to open in the design area. Perform the following steps:

1. In the **Widgets** pane, drag the **Label** widget and place it at coordinates (0,13) and do the these steps:

    1. Type `Welcome, email@test.com`.
    2. In the **Widget Interactions and Notes** pane, click in the **Shape Name** field and type `WelcomeLabel`.
    3. In the **Widget Properties and Style** pane, with the **Style** tab selected scroll to **Font**, change the font size to `9`, and click on the Italic icon *I*.

2. In the **Widgets** pane, drag the **Button Shape** widget and place it at coordinates (164,10). Type `Log Out`. In the toolbar, change **w:** to `56` and **h:** to `16`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `LogOutButton`.

To complete the design of the **Header** master, we need to rename the menu items on the **HzMenu**. In the **Masters** pane, double-click on the **Header** master to open in the design area. Click on the **HzMenu** at coordinates (250,80). Perform the following steps:

1. Click on the first menu item and type `Random Musings`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `RandomMusingsMenuItem`. Click on **Case 1** under the **OnClick** event and press the *Delete* key. Click on **Create Link…**. In the pop-up sitemap, click on **Random Musings**.
2. Again, click on the first menu item and type `Accolades and News`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `AccoladesMenuItem`. Click on **Case 1** under the **OnClick** event and press the *Delete* key. Click on **Create Link…**. In the pop-up sitemap, click on **Accolades and News**.
3. Click on the first menu item and type `About`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type

`AboutMenuItem`. Click on **Case 1** under the **OnClick** event and press the *Delete* key. Click on **Create Link…**. In the pop-up sitemap, click on **About**.

We will now create a registration lightbox that will be shown when the user clicks on the **NewUserLink**.

Note

To display a dynamic panel in a lightbox, we will use the **OnShow** action with the option **treat as lightbox** set. We will use the **Registration** dynamic panel's **Pin to Browser** property to have the dynamic panel shown in the center and middle of the window. Learn more at http://www.axure.com/learn/dynamic-panels/basic/lightbox-tutorial.

1. In the **Masters** pane, double-click on the icon next to the **Header** master to open in the design area. In the **Widgets** pane, drag the **Dynamic Panel** widget and place it at coordinates (310,200). In the toolbar, change **w:** to `250`, **h:** to `250`, and click on the **Hidden** checkbox. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `RegistrationLightBoxDP`. In the **Widget Manager** pane with the **Properties** tab selected, click on **Pin to Browser**. In the **Pin to Browser** dialog box, click on the checkbox next to **Pin to browser window** and click on **OK**.
2. In the **Widget Manager** pane, under the **RegistrationLightBoxDP**, click on the State icon beside **State1** twice to open in the design area.
3. In the **Widgets**pane, drag the **Rectangle** widget and place it at coordinates (0,0). In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `BackgroundRectangle`. In the toolbar, change **w:** to `250` and **h:** to `250`.
4. Again in the **Widgets** pane, drag the **Heading2** widget H2 and place it at coordinates (25,20).
5. With the **Heading2** widget selected, type `Registration`. In the toolbar, change **w:** to `141` and **h:** to `28`.
6. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `RegistrationHeading`.
7. Repeat steps 8-10 to complete the design of the **RegistrationLightBoxDP** using the following table (* if applicable):

Widget	Coordinates	Text* (Shown on Widget)	Width* (w:)	Height* (h:)	Name field (In the Widget Interactions and Notes pane)
 Label	(25,67)	`Enter Email`			**EnterEmailLabel**
 Text Field	(25,86)				**EnterEmailField**
 Label	(25,121)	`Enter Password`			**EnterPasswordLabel**
 Text Field	(25,140)				**EnterPasswordField**
 Button Shape	(25,190)	`Submit`	200	30	**SubmitButton**

8. Click on the **EnterEmailField** text field at coordinates (25,86). In the **Widget Properties and Style** pane, with the **Properties** tab selected, scroll to **Text Field** and perform the following steps:

    1. Next to**Hint Text**, enter `Email`. Click on **Hint Style**. In the **Set Interaction Styles** dialog box, click on the checkbox next to **Font**

Color. Click on the down arrow next to the Text Color icon . In the drop-down menu, in the **#** text field, enter `999999`.
    2. Click on **OK**.
9. Click on the **EnterPasswordField** text field at coordinates (25,140). In the **Widget Properties and Style** pane, with the **Properties** tab selected, scroll to **Text Field** and perform the following steps:

    1. Click on the drop-down menu next to **Type** and select `Password`.
    2. Next to **Hint Text**, enter `Password`.
    3. Click on **Hint Style**. In the **Set Interaction Styles** dialog box, click on the checkbox next to **Font Color**. Click on the down arrow next to the Text Color icon . In the drop-down menu, in the **#** text field, enter `999999`.
    4. Click on **OK**.

10. With the updates completed for the **Header** master, we are now ready to define the interactions.

Refining the interactions for our Header master

We will need to add additional interactions for **Log In** and **Registration** on our **Header** master.

Interactions with our **Header** master will be triggered by the following named widgets and events:

Dynamic Panel	State	Widget	Event
LoginDP	**State1**	**LoginButton**	**OnClick**
LoginDP	**State1**	**NewUserLink**	**OnClick**
LoginDP	**State1**	**ForgotLink**	**OnClick**

Dynamic Panel	State	Widget	Event
LoginDP	State2	LogOutButton	OnClick
RegistrationLightBoxDP	State1	SubmitButton	OnClick

We will now define the interactions for each widget, starting with **LoginButton**.

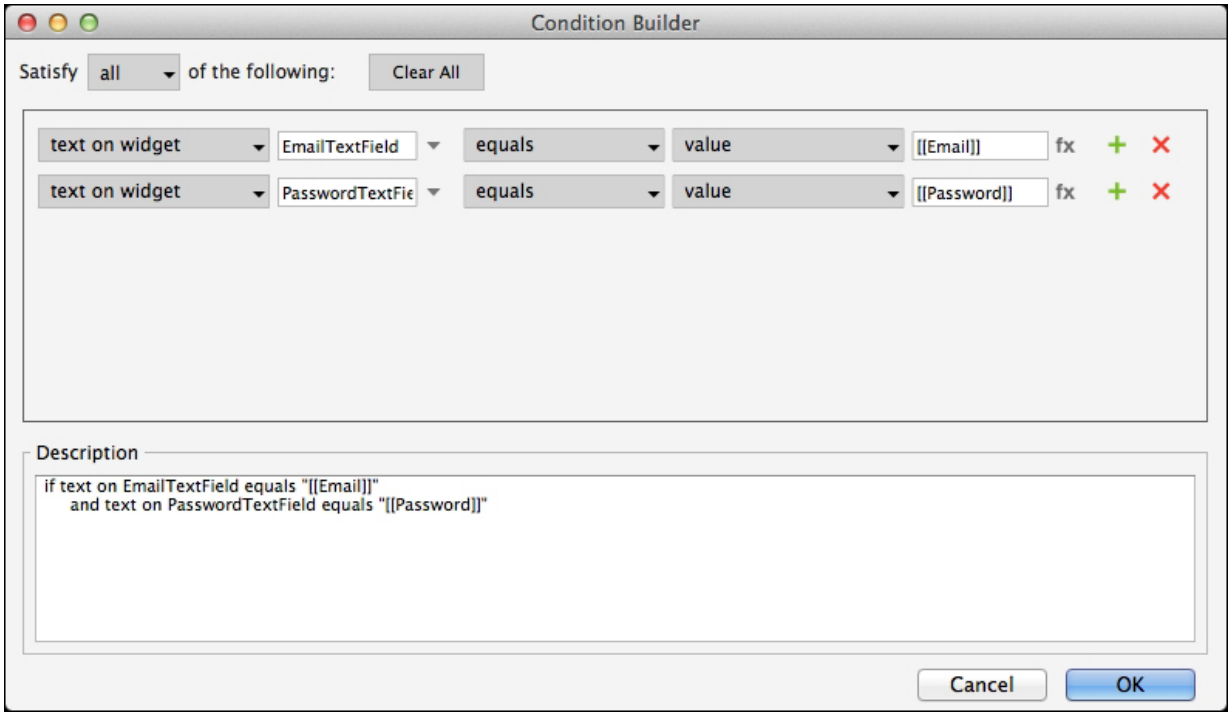
Defining interactions for the LoginButton

When the **LoginButton** is clicked, the **OnClick** event will evaluate if the text entered in the **EmailTextField** and **PasswordTextField** equals the e-mail and password variable values. If the variables are valid, **LoginDP** will be set to **State2** and text on the **WelcomeLabel** will be updated. If the variables values are not equal, we will show an error message. We will define these actions by creating two cases: **ValidateUser** and **ShowErrorMessage**.

Validating the user's email and password

To define the **ValidateUser** case for the **OnClick** interaction, open the **LogInDP State1** in the design area. Click on the **LogInButton** at coordinates (164,10). In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case Name** field, type `ValidateUser`. In the **Case Editor** dialog, perform the following steps:

You will see the **Condition Builder** window similar to the one shown in the following screenshot after the first and second conditions are defined:



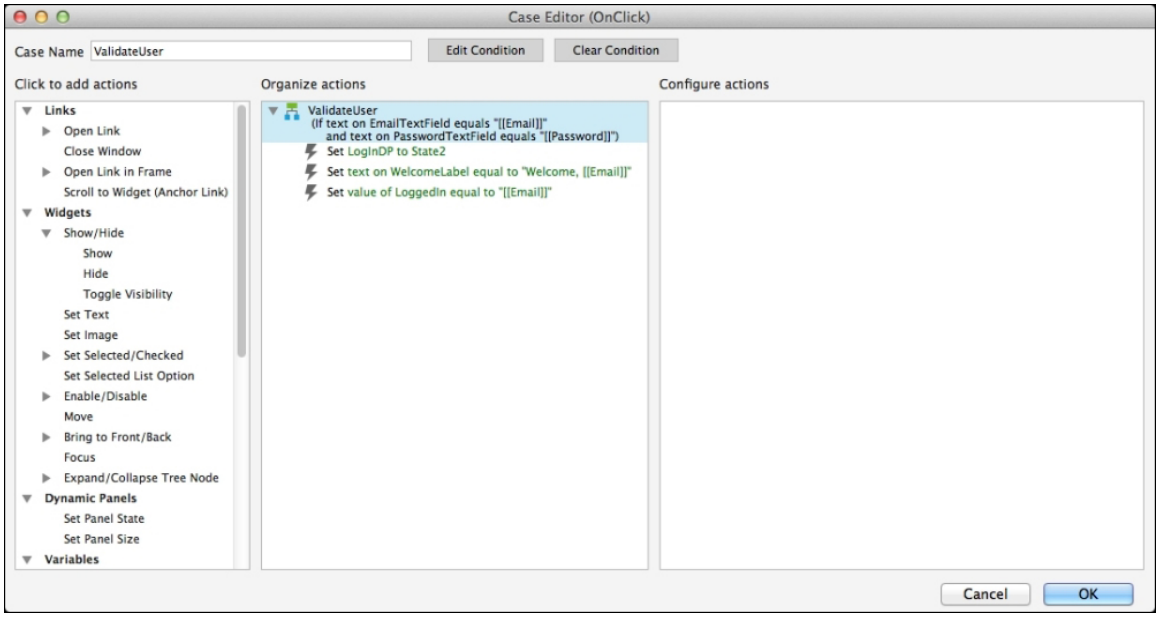
1. Create the first condition. Click on the **Add Condition** button.
2. In the **Condition Builder** dialog box, in the outlined condition box, perform the following steps:

    1. In the first dropdown, select **text on widget**.
    2. In the second dropdown, select **EmailTextField**.
    3. In the third dropdown, select **equals**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth dropdown, select **[[Email]]**.
    6. Click the green **+** sign.

3. Create the second condition. Click on the **Add Condition** button.
4. In the **Condition Builder** dialog box, in the outlined condition box, perform the following steps:

    1. In the first dropdown, select **text on widget**.
    2. In the second dropdown, select **PasswordTextField**.
    3. In the third dropdown, select **equals**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth dropdown, select **[[Password]]**.
    6. Click on **OK**.

5. Once the following three actions are defined, you should see the **Case Editor** similar to the one shown in the following screenshot:



6. Create the first action. To set panel state for the **LogInDP** dynamic panel, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Dynamic Panels** drop-down menu and click on **Set Panel State**.
    2. Under **Configure actions**, click on the checkbox next to **LoginDP**.
    3. Next to **Select the state**, click on the dropdown and select **State2**.

7. Create the second action. To set text for the **WelcomeLabel**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click the checkbox next to **WelcomeLabel**.
    3. Under **Set text to**, click on the dropdown and select **value**. In the text field, enter `Welcome, [[Email]]`.

8. Create the third action. To set value of the **LoggedIn** variable, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **LoggedIn**.
    3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `[[Email]]`.
    4. Click on **OK**.

With the **ValidateUser** case completed, next we will create the **ShowErrorMessage** case.

Creating the ShowErrorMessage case

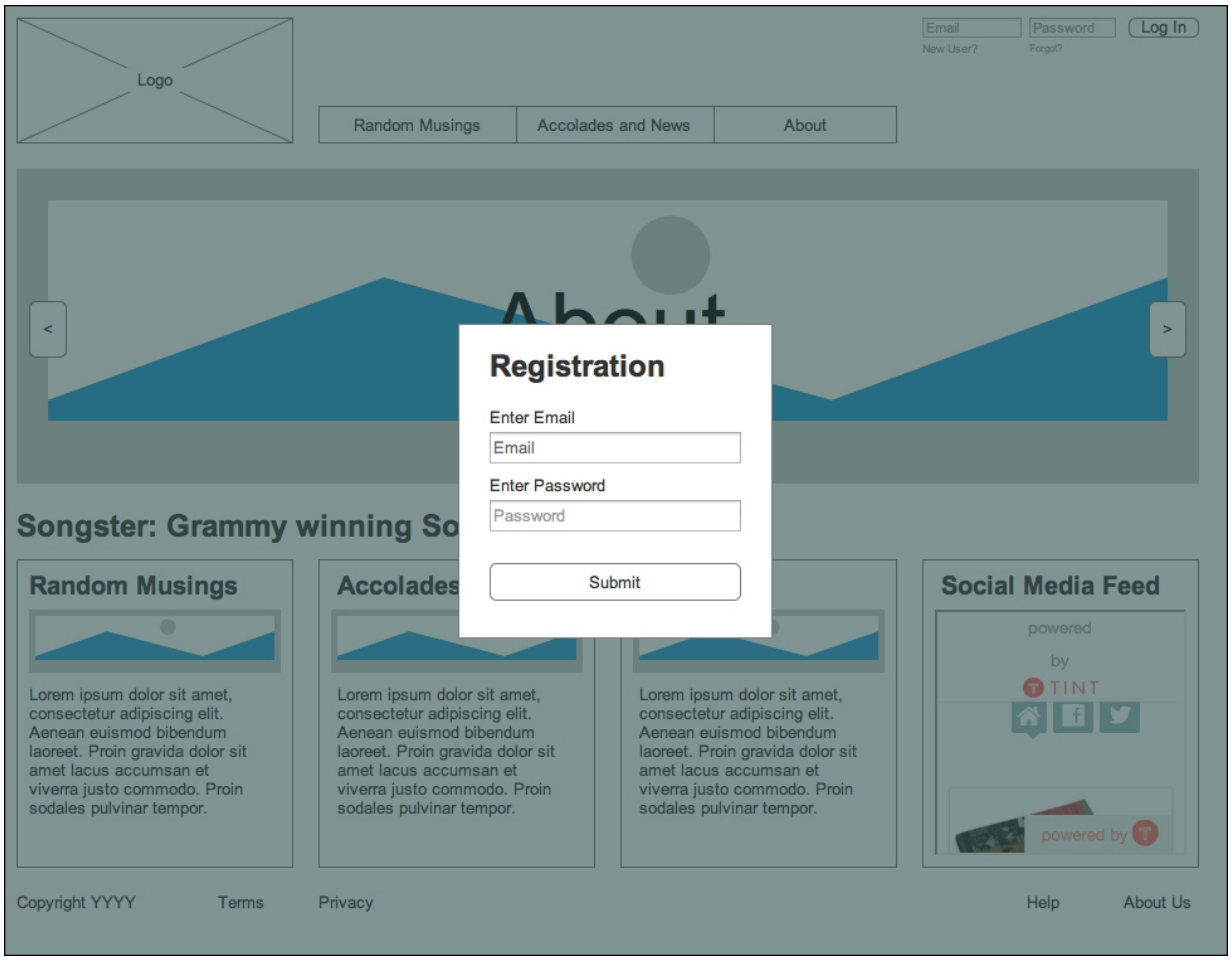
To create the **ShowErrorMessage** case, in the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case Name** field, type `ShowErrorMessage`. Create the action. To show the **ErrorMessage** label, perform the following steps:

1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Show**.
2. Under **Configure actions**, under **LoginDP** dynamic panel, click on the checkbox next to **ErrorMessage**.
3. Click on **OK**.

Next, we will enable the interaction for the **NewUserLink**.

Enabling interaction for the NewUserLink

When the **NewUserLink** is clicked, the **OnClick** event will show the **RegistrationLightBox** dynamic panel as a lightbox, as shown in the following screenshot:



With the **LogInDP State1** still opened in the design area, click on the **NewUserLink** at coordinates (0,30). To enable the **OnClick** event in the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case Name** field, type `ShowLightBox`. Now, create the action; to show the **RegistrationLightBox**, perform the following steps:

1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown, and click on **Show**.
2. Under **Configure actions**, click on the checkbox next to **RegistrationLightBoxDP**.
3. Next go to **More options**, click on the dropdown and select **treat as lightbox**.
4. Click on **OK**.

Next, we will activate interactions for the **ForgotLink**.

Activating interactions for the ForgotLink

When the **ForgotLink** is clicked, the **OnClick** event will show the **RegistrationLightBox** dynamic panel as a lightbox, the **RegistrationHeading** text will be updated to display **Forgot Password?** and the **EnterPassworIdLabel,** as well as the **EnterPasswordField,** will be hidden.

To enable the **OnClick** event, in the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case Name** field, type `ShowForgotLB`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action; to show the **RegistrationLightBox**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Show**.
    2. Under **Configure actions**, click on the checkbox next to **RegistrationLightBoxDP**.
    3. Next, go to **More options**, click on the dropdown and select **treat as lightbox**.

2. Create the second action; to set text for the **RegistrationHeading**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **RegistrationHeading**.
    3. Under **Set text to**, click on the dropdown and select **value**. In the text field, enter `Forgot Password?`.

3. Create the third action; to hide the **EnterPasswordLabel** and **EnterPasswordField**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown, and click on **Hide**.
    2. Under **Configure actions**, under **RegistrationLightBoxDP**, click on the checkboxes next to **EnterPasswordLabel** and **EnterPasswordField**.
    3. Click on **OK**.

We have now completed the interactions for **State1** of **LoginDP**. Next, we will facilitate interactions for the **LogOutButton**.

Facilitating interactions for the LogOutButton

When the **LogOutButton** is clicked, the **OnClick** event will perform the following actions:

- Hide the **ErrorMessage** on the **LoginDP State1**
- Set text for **PasswordTextField** and **EmailTextField**
- Set panel state for **LoginDP** to **State1**
- Set variable value for **LoggedIn**

To enable the **OnClick** event, open the **LogInDP State2** in the design area. Click on the **LogInOut** at coordinates (164,10). In the **Widget Interactions and Notes** pane, with the **Interactions** tab selected, click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case Name** field, type `LogOut`. In the **Case Editor** dialog, perform the following steps:

1. Create the first action; to hide the **ErrorMessage**, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown, and click on **Hide**.
    2. Under **Configure actions**, under **LoginDP**, click on the checkbox next to **ErrorMessage**.
2. Create the second action; to set text for the **PasswordTextField** and **EmailTextField**, perform the following steps:
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click the checkbox next to **PasswordTextField**.
    3. Under **Set text to**, click the dropdown and select **value**. In the text field, clear any text shown.
    4. Under **Configure actions**, click the checkbox next to **EmailTextField**.
    5. Under **Set text to**, click on the dropdown and select **value**. In the text field, enter `Email`.
3. Create the third action; to set panel state for the **LogInDP** dynamic panel, perform the following steps:

1. Under **Click to add actions**, scroll to the **Dynamic Panels** drop-down menu and click on **Set Panel State**.
2. Under **Configure actions**, click on the checkbox next to **LoginDP**.
3. Next to **Select the state**, click on the dropdown and select **State1**.

4. Create the fourth action. To set variable value of LoggedIn, perform the following steps:

1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
2. Under **Configure actions**, click on the checkbox next to **LoggedIn**.
3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `No`.
4. Click on **OK**.

We have now completed interactions for **State2** of the **LoginDP**. Next, we will construct interactions for the **RegistrationLightBoxDP**.

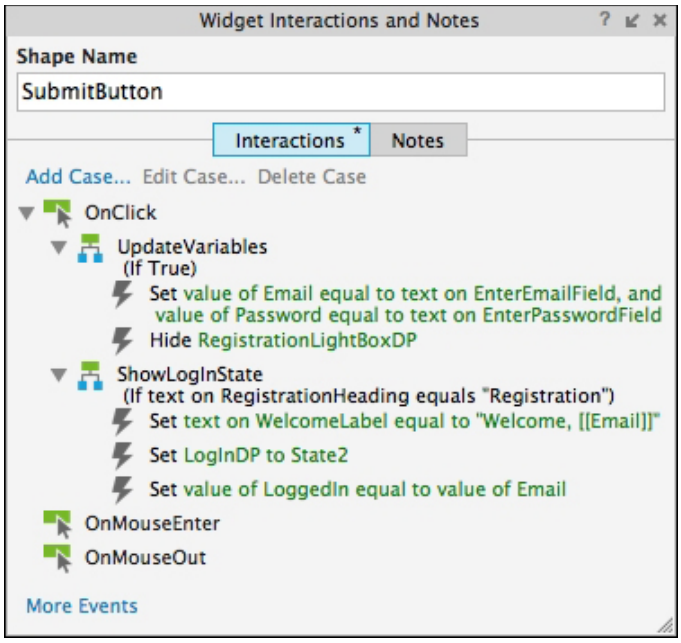
Constructing interactions for the RegistrationLightBoxDP

When the **LoginButton** is clicked, the **OnClick** event hides **RegistrationLightBoxDp** and sets the **Email** and **Password** variable values to the text entered in the **EnterEmailField** and **EnterPasswordField**. Also, if text on the **RegistrationHeading** label is equal to **Registration**, **LoginDP** will be set to **State2**. We will define these actions by creating two cases: **UpdateVariables** and **ShowLogInState**.

Updating Variables and hiding the RegistrationLightBoxDP

In the **Widget Manger** pane, double-click on the **RegistrationLightBoxDP State1** to open in the design area. To define the **UpdateVariables** case for the **OnClick** interaction, click on the **SubmitButton** at coordinates (25,190). In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case Name** field, type `UpdateVariables`. In the **Case Editor** dialog, perform the following steps:

The following screenshot shows **Case Editor** with the actions defined:



1. Create the first action; to set variable value for the **Email** and **Password** variables, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **Email**.
    3. Under **Set variable to**, click on the first dropdown and select **text on widget**. Click on the second dropdown and select **EnterEmailField**.
    4. Under **Configure actions**, click on the checkbox next to **Password**.
    5. Under **Set variable to**, click on the first dropdown and select **text on widget**. Click on the second dropdown and select **EnterPasswordField**.

2. Create the second action; to hide **RegistrationLightBoxDP**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Hide**.
    2. Under **Configure actions**, click on the checkbox next to **RegistrationLightBoxDP**.
    3. Click on **OK**.

With the **UpdateVariables** case completed, next we will create the **ShowLogInState** case.

Creating the ShowLoginState case

To create the **ShowLogInState** case, in the **Widget Interactions and Notes** pane with the **Interactions** tab selected click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case Name** field, type `ShowLogInState`. In the **Case Editor** dialog, perform the following steps:

1. Click on the **Add Condition** button to create the first condition.
2. In the **Condition Builder** dialog box, go to the outlined condition box and perform the following steps:

    1. In the first dropdown, select **text on widget**.
    2. In the second dropdown, select **RegistrationHeadline**.
    3. In the third dropdown, select **equals**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth dropdown, select **Registration**.
    6. Click on **OK**.

3. Create the first action; to set text for the **WelcomeLabel**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **WelcomeLabel**.
    3. Under **Set text to**, click on the dropdown and select **value**. In the text field, enter `Welcome, [[Email]]`.
    4. Click on **OK**.

4. Create the second action; to set panel state for the **LogInDP** dynamic panel, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Dynamic Panels** drop-down menu and click on **Set Panel State**.
    2. Under **Configure actions**, click on the checkbox next to **LoginDP**.
    3. Next to **Select the state**, click on the dropdown and select **State2**.

5. Create the third action; to set value of the **LoggedIn** variable, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **LoggedIn**.

3. Under **Set variable to**, click on the first dropdown and click on **value**. In the text field, enter `[[Email]]`.
4. Click on **OK**.

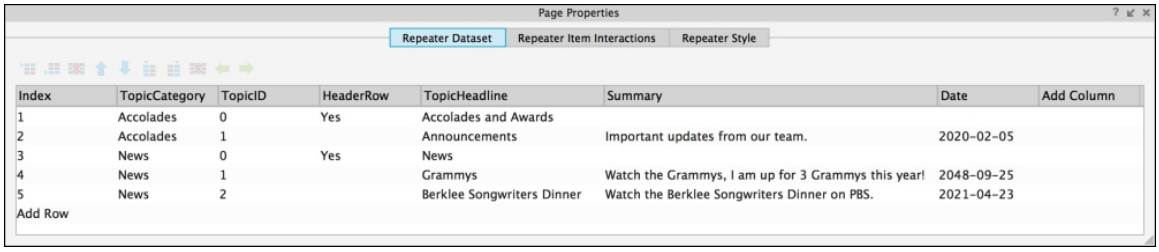
6. Under the **OnClick** event, right-click on the **ShowErrorMessage** case and click on **Toggle IF/ELSE IF**.

With our **Header** master updated, we are now ready to refresh data for our **Forum** repeater.

Refreshing data for our Forum repeater

Recall that the **ForumRepeater** has a total of seven columns and five rows of data. We will need to refresh the **Repeater Dataset** for our project. Perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Forum** master to open in the design area.
2. Double-click on the **ForumRepeater** widget to open in the design area.
3. Under the design area in the **Repeater** pane, click the **Repeater Dataset** tab. Use the following screenshot as a reference to update the **Repeater Dataset**:

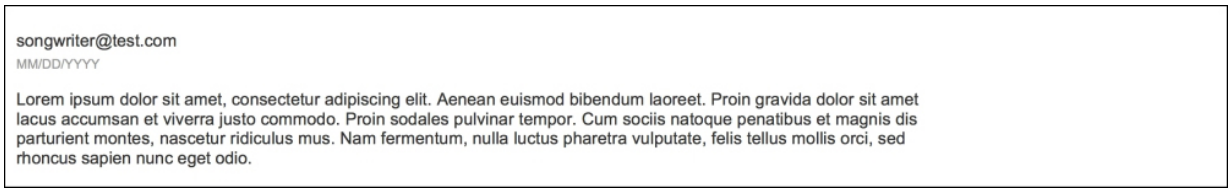


With the **Repeater Dataset** refreshed for our **Forum** repeater, we are now ready to assemble our **Post Commentary** master.

Assembling our Post Commentary master

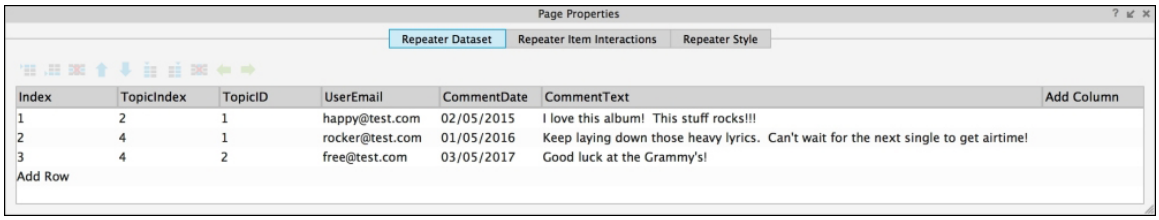
Recall from our user stories that the songwriter would like to enable users to add commentary to the posts. To accomplish this, we will need to design a **Post Commentary** master that will be added to our **Topic Detail** page.

Once completed, the default repeater item on the **Post Commentary** master will look as follows:



The design for our **Post Commentary** master comprises a single Repeater. To update the **Post Commentary** master, perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **PostCommentary** master to open in the design area.
2. In the **Widgets** pane, click on the **Repeater** widget . Drag the **Repeater** widget onto the design area and place at coordinates (0,0). In the **Widget Interactions and Notes** pane, click on the **Repeater Name** field and type `PostCommentaryRepeater`.
3. Double-click on the **PostCommentaryRepeater** widget to open in the design area.
4. Under the design area in the **Repeater** pane, click on the **Repeater Style** tab. To change the default repeater style, under the **Layout** dropdown, click on the radio button next to **Horizontal** and then click the checkbox next to the **Wrap (Grid)** field. In the **Items per row** field, enter 1.
5. Under the design area in the **Repeater** pane, click on the **Repeater Dataset** tab. The **PostCommentaryRepeater** will have a total of six columns and three rows of data. Once completed, our **Repeater Dataset** will look as follows:



Note

Use the preceding screenshot as a reference to populate the **Repeater Dataset**. Click on **Add Column** and **Add Row** to add additional columns and rows, as needed. **TopicIndex** and **TopicID**

cross reference to the Forum **Repeater Dataset** columns, **Index** and **TopicID** respectively.

6. To update and place widgets on the **Post Commentary** repeater, double-click on the **PostCommentaryRepeater** to open in the design area. Click on the **Rectangle** widget at coordinates (0,0). In the toolbar, change **w:** to `940` and **h:** to `140`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `PCBackground`.
7. In the **Widgets** pane, drag the **Label** widget  and place it at coordinates (10,20). In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `UserEmail`.
8. In the **Widgets** pane, drag the **Label** widget  and place it at coordinates (10,40).
9. Select the **Label** widget.
10. Type `MM/DD/YYYY`. In the toolbar, change **w:** to `65` and **h:** to `11`.
11. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `CommentDate`.
12. In the **Widget Properties and Style** pane, with the **Style** tab selected, scroll to **Font** and perform the following steps:

    1. Change the font size to `10`.
    2. Click on the down arrow next to the Text Color icon . In the dropdown menu, in the **#** text field, enter `999999`.
    3. Under the **Alignment + Padding** dropdown, click on the Align Left icon .

13. In the **Widgets** pane, drag the **Paragraph** widget  and place it at coordinates (10,65). In the toolbar, change **w:** to `700` and **h:** to `60`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `CommentText`.

Next, we will define the **OnItemLoad** event actions for the **PostCommentary** repeater.

Defining the OnItemLoad event actions

Double-click on the icon  next to the **PostCommentary** master to open in the design area. Click on the **PostCommentary** repeater at coordinates (0,0). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, then click on **More Events**, and then double-click on **OnItemLoad**. A

Case Editor dialog box will open. In the **Case Editor** dialog, perform the following steps:

1. In the **Case Name** field, type `SetPostCommentary`.
2. Create the action; to set text on **UserEmail**, **CommentDate**, and **CommentText**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **UserEmail**. Under **Set text to**, click on the first dropdown and click on **value**. In the text field, enter `[[Item.UserEmail]]`.
    3. Under **Configure actions**, click on the checkbox next to **CommentDate**. Under **Set text to**, click on the first dropdown and click on **value**. In the text field, enter `[[Item.CommentDate]]`.
    4. Under **Configure actions**, click on the checkbox next to **CommentText**. Under **Set text to**, click on the first dropdown and click on **value**. In the text field, enter `[[Item.CommentText]]`.

3. With the design and interactions completed for the **PostCommentary** repeater, we are now ready to manufacture a **NewPost** master.

Manufacturing a NewPost master

Our new post master will display a link enabling authorized users to create new posts. When an authorized user clicks on the link, a **NewPost** lightbox will be displayed. Submitting the new post will add a row to the **Forum** repeater.

1. To manufacture the new post master; in the **Masters** pane, double-click on the icon  next to the **NewPost** master to open in the design area.

    In the **Widgets** pane, drag the **Dynamic Panel** widget  and place it at coordinates (310,200). In the toolbar, change **w:** to `250`, **h:** to `250` and click on the **Hidden** checkbox. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `NewPostLightBoxDP`. In the **Widget Manager** pane with the **Properties** tab selected, click on **Pin to Browser**. In the **Pin to Browser** dialog box, click on the checkbox next to **Pin to browser window** and click on **OK**.
2. In the **Widget Manager** pane, under **NewPostLightBoxDP**, click on the State icon  beside **State1** twice to open in the design area. Perform the following steps:

1. In the **Widgets** pane, drag the **Rectangle** widget  and place it at coordinates (0,0). In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `BackgroundRectangle`. In the toolbar, change **w:** to `250` and **h:** to `250`.
2. In the **Widgets** pane, drag the **Heading2** widget  and place it at coordinates (25,20). Type `Add Thought`. In the toolbar, change **w:** to `150` and **h:** to `28`.

3. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `NewPostHeading`.
4. Repeat step 2 to complete the design of **RegistrationLightBoxDP** using the following table (* if applicable):

Widget	**Coordinates**	**Text*** **(Shown on Widget)**	**Width*** **(w:)**	**Height*** **(h:)**	**Name field** **(In the Widget Interactions and Notes pane)**
 Text Area	(25,60)				**NewPostTextArea**
 Button Shape	(25,190)	`Submit`	`200`	`30`	**NewPostSubmitButton**

5. To create the first action, raise event **OnClickSubmit**, perform the following steps:

    1. Click on the **SubmitButton** at coordinates (25,190). In the **Widget Interactions and Notes** pane, double-click on **OnClick**.
    2. Under **Click to add actions**, scroll to the **Miscellaneous** drop-down menu and click on **Raise Event**.

3. Under **Configure actions**, click on the green **+** and type `OnClickSubmit`. Click on the checkbox next to **OnClickSubmit**.

6. Create the second action; to set value of the **UserText** variable, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **UserText**.
    3. Under **Set variable to**, click on the first dropdown and click on **text on widget**. In the text field dropdown, select **NewPostTextArea**.
    4. Click on **OK**.

We now need to duplicate **State1**. We will then modify the new **State2** enabling new post creation. In the **Widget Manager** pane, under **NewPostLightBoxDP**, right-click on **State1** and in the menu, click on **Duplicate State**. Click on the State icon beside **State2** twice to open in the design area. Perform the following steps:

1. Click on the **NewPostHeading** at coordinates (25,20) and move to coordinates (25,0). Type `Create New Post`.
2. Click on the **NewPostTextArea** at coordinates (25,60) and move to coordinates (25,97).
3. Click on the **NewPostSubmitButton** at coordinates (25,190) and move to coordinates (25,210).
4. In the **Widgets** pane, drag the **Droplist** widget  and place it at coordinates (25,0) and perform the following steps:
5. Right-click on the **Droplist** and click on **Edit List Items**. Click on the green **+** sign and type `Accolades and Awards`. Click on the green **+** sign and type `News`. Click on **OK**.
6. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `CategoryDroplist`.
7. In the **Widgets** pane, drag the **Text Field** widget  and place it at coordinates (25,64). In the **Widget Properties and Style** pane, with the **Properties** tab selected scroll to **Text Field**, perform the following steps:

    1. Next to **Hint Text**, enter `Headline`.
    2. Click on **Hint Style**. In the **Set Interaction Styles** dialog box, click on the checkbox next to **Font Color**. Click on the down arrow next to the Text Color icon . In the drop-down menu, in the **#** text field, enter `999999`.

8. In the **Widget Interactions and Notes** pane, under **Case 1**, double-click on the **Set value** action. Perform the following steps:

    1. Under **Configure actions**, click on the checkbox next to **NewPostTopic**.
    2. Under **Set variable to**, click on the first dropdown and click on **selected option of**. In the text field dropdown, select **CategoryDroplist**.
    3. Under **Configure actions**, click on the checkbox next to **NewPostHeadline**.
    4. Under **Set variable to**, click on the first dropdown and click on **selected option of**. In the text field dropdown, select **Headline**.
    5. Click on **OK**.

With the design and interactions completed for the **NewPost** master, we are now ready to repurpose all pages in our sitemap.

Repurposing pages in our sitemap

We will now update all pages in our **Sitemap**. The client has told us that BreadCrumbs should not be used on the site. We will remove the **BreadCrumb** master from each page in the **Sitemap** pane. Also, we will update the **OnPageLoad** event for each page refreshing the **Headline** text, as well as removing actions that update **BreadCrumb**. Additional enhancements will be made to the **Random Musings**, **Topic List**, and **Topic Detail** pages to enable post creation and commentary.

Remodeling our home page

From an architectural perspective, our home page layout remains similar to the community site home page. As such, we need to update our carousel, headline, and our call-to-action boxes as follows:

1. In the **Sitemap** pane, double-click on the Page icon next to the **Home** page to open in the design area.
2. In the **Widget Manager** pane, click on the State icon beside **State1** twice to open in the design area. Click on the Image at coordinates (0,0). Type `Accolades and News`.
3. In the **Widget Manager** pane, click on the State icon beside **State2** twice to open in the design area. Click on the Image at coordinates (0,0). Type `Random Musings`.
4. In the **Widget Manager** pane, click on the State icon beside **State3** twice to open in the design area. Click on the Image at coordinates (0,0). Type `About`.
5. Click on the **Heading 1** widget at coordinates (10,400). Type `Songster: Grammy Winning Songwriter`.
6. Click on the **Heading 2** widget at coordinates (20,450). Type `Random Musings`.

> **Note**
>
> You will have to click twice to select these widgets, since the **Heading 2** widget is behind a **Section Heading** Hot Spot widget.

7. Click on the **Heading 2** widget at coordinates (265,450). Type `Accolades and News`.
8. Click on the **Heading 2** widget at coordinates (505,450). Type `About`.

With our home page ready, we can now focus on our remaining pages.

Invigorating our remaining pages

We will now update the remaining pages in our sitemap. We will refresh the **Header** name, delete the **BreadCrumb** master, as well as update the **OnPageLoad Initialize** case on each page. To update the pages, perform the following steps:

1. In the **Sitemap** pane, double-click on the Page icon next to the **Accolades and News** page to open in the design area.

2. Click on the **Header** at coordinates (10,10). In the **Widget Interactions and Notes** pane, click in the **Header Name** field and type `Accolades_Page_Header`.
3. Click on the **BreadCrumb** master at coordinates (10,113) and press *Delete*.
4. In the **Page Properties** pane under the design area, click on the **Page Interactions** tab. Under the **OnPageLoad** event under the **Initialize** case, perform the following steps:

    1. Click on the **Set value of HorizontalOffset equal to "0"** action and press *Delete*. Click on the **Add Rows** last action and press *Delete*.
    2. Double-click on the **Set text** case. In **Case Editor**, under **Configure actions** in the **Set text to** field, enter `Accolades and News`.

5. In the **Page Properties** pane under the design area, click on the **Page Interactions** tab. Under the **OnPageLoad** event under the **Initialize** case, perform the following steps:

    1. Double-click on the **Set is selected …** case. In the **Case Editor** window, navigate to **Configure actions** | **Select the widgets to set selected state** and verify that the appropriate menu item is selected.

    > ## Note
    >
    > For example, for the **Accolades and News** page under the **Accolades_Page_header (header)** dropdown, the checkbox would be checked next to **AccoladesMenuItem (Menu Item)**.

    2. Click on **OK**.

6. Repeat steps 1-5 to update the remaining pages using the values given in the following table (*step not applicable for rows with no data):

Page to Open (Step 1)	**Header Name (Step 2)**	**Set Text to* (Step 4)**	**Set is selected* (Step 5)**
Topic List	**Topic_List_Page_Header**		**Topic_List_Page_Header** **RandomMusingsMenuItem**
Topic Detail	**Topic_Detail_Page_Header**	`[[TopicHeadline]]`	**Topic_List_Page_Header** **RandomMusingsMenuItem**

Page to Open (Step 1)	Header Name (Step 2)	Set Text to* (Step 4)	Set is selected* (Step 5)
About	**About_Page_Header**	`About`	
Terms	**Terms_Page_Header**	`Terms`	
Privacy	**Privacy_Page_Header**	`Privacy`	
Contact	**Contact_Page_Header**	`Contact`	
Random Musings	**R_M_Page_Header**		**Topic_List_Page_Header** **RandomMusingsMenuItem**

To complete the **Random Musings** page, with the page still opened in the design area click on the **Headline** widget at (20,147) and type `Random Musings`.

With the interior pages completed, we are ready to enable post and commentary creation for the **Topic Detail**, **Random Musings**, and **Topic List** pages.

Enabling commentary and new post creation

We will first enable commentary on the **Topic Detail** page by adding the **PostCommentary** master and refining interactions. Next, we will add the **NewPost** master to the **Random Musings** and **Topic List** pages, permitting the creation of new posts for authorized users.

Facilitating commentary creation

To facilitate commentary creation on the **Topic Detail** page, we will need to place an **Add Your Thoughts** link as well as the **PostCommentary** master. When the user clicks on the link, the **PostCommentary** master will be shown. The user can then enter their comment. When the user clicks on the **Submit** button, the comment is added to the **PostCommentaryRepeater**.

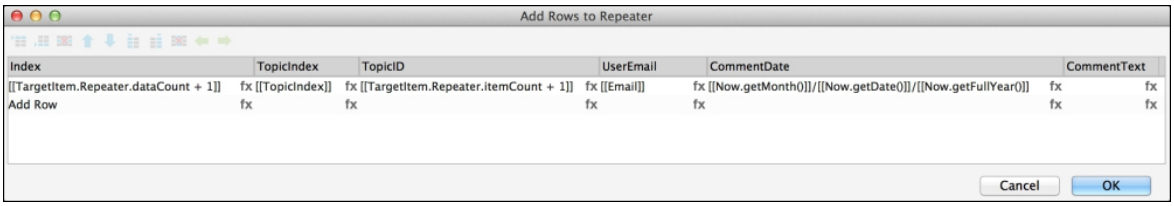
To update the **Topic Detail** page, in the **Sitemap** pane, double-click on the Page icon next to the **Topic Detail** page to open in the design area. Perform the following steps:

1. Right-click on the footer at coordinates (10,715) and click on **Break Away**. In the toolbar, change **y:** to `1685`. All widgets of the footer should move to the new *y* coordinate.

2. In the **Masters** pane, drag the **PostCommentary** master and drop it at coordinates (10,760) in the design area. In the **Widget Interactions and Notes** pane, click on the **PostCommentary Name** field and type `CommentaryThread`.
3. In the **Masters** pane, drag the **NewPost** master and drop it anywhere in the design area. In the **Widget Interactions and Notes** pane, perform the following steps:

    1. Click on the **NewPost Name** field and type `NewComment`.
    2. To create the **Add Row** action, double-click on **OnClickSubmit**. In the **Case Editor** dialog, perform the following steps:

4. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click on **Add Rows**.
5. Under **Configure actions**, click on the checkbox next to **PostCommentaryRepeater**.
6. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update row values for all columns using the following screenshot:



7. Create the **Hide the NewPostLightBoxDP** action, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Hide**.
    2. Under **Configure actions** and then under **Select the widget to hide/show** in the text field, type `NewPost`.
    3. Under **NewComment**, click on the checkbox next to **NewPostLightBoxDP**.
    4. Click on **OK**.

8. In the **Widgets** pane, drag the **Heading 2** widget H2 and place it at coordinates (10,712). Type `Add Your Thoughts`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `AddThoughtsLink`. In the **Widget Properties and Style** pane, with the **Style** tab selected, scroll to **Font** and perform the following steps:

    1. Click on the Underline icon U.
    2. Click on the down arrow next to the Text Color icon A. In the dropdown menu, in the **#** text field, enter `0033FF`.

9. In the **Widgets** pane, drag the **Label** widget A_ and place it at coordinates (310,717).
10. Select the **Label** widget and perform the following steps:
11. Type `Error: Please Log In to Add Your Thoughts.`.
12. In the **Widget Interactions and Notes** pane, click in the **Shape Name** field and type `ThoughtErrorMessage`. In the toolbar, click on the checkbox next to **Hidden**.
13. In the **Widget Properties and Style** pane, with the **Style** tab selected, scroll to **Font** and perform these steps:

    1. Change the font size to `16`.

2. Click on the down arrow next to the Text Color icon . In the dropdown menu, in the **#** text field, enter `FF0000`.
3. In the toolbar, click on the checkbox next to **Hidden**.

We will now focus on creating the **UserLoggedIn** case for the **OnClick** event on the **AddThoughtsLink**. Click on the **AddThoughtsLink** at coordinates (10,712) to select. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case Name** field, type `UserLoggedIn`. In the **Case Editor** dialog, perform the following steps:

1. Click on the **Add Condition** button to create the condition. In the **Condition Builder** dialog box, in the outlined condition box, perform the following steps:

    1. In the first dropdown, select **value of variable**.
    2. In the second dropdown, select **LoggedIn**.
    3. In the third dropdown, select **does not equal**.
    4. In the fourth dropdown, select **value**.
    5. In the text field type **No**.
    6. Click on **OK**.

2. Create the first action; to show **NewPostLightBoxDP**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Show**.
    2. Under **Configure actions**, under **Select the widget to hide/show** in the text field, type `NewPost`.
    3. Under **NewComment**, click on the checkbox next to **NewPostLightBoxDP**.
    4. Next to **More options**, click on the dropdown and select **treat as lightbox**.
    5. Under **NewComment**, under **NewPostLightBoxDP,** click on the checkboxes next to **BackgroundRectangle**, **NewPostHeading**, **NewPostTextArea**, and **NewPostSubmitButton**.

3. Create the second action; to hide **ThoughtErrorMessage**, perform the following steps:

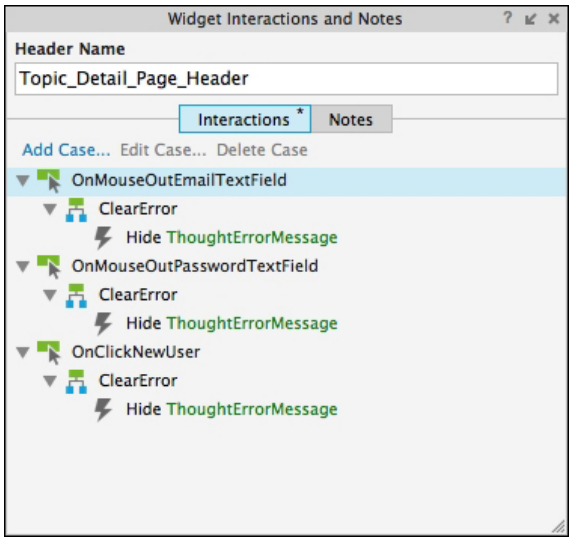
    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Hide**.
    2. Under **Configure actions**, under **Select the widget to hide/show** in the text field, type `Thought`.
    3. Click on the checkbox next to **ThoughtErrorMessage**.
    4. Click on **OK**.

We will now focus on creating the **NotLoggedIn** case for the **OnClick** event on the **AddThoughtsLink**. With the **AddThoughtsLink** still selected, in the **Widget Interactions and Notes** pane, click on the **Interactions** tab and click on **Add Case…**. A **Case Editor** dialog box will open. In the **Case Name** field, type `NotLoggedIn`. In the **Case Editor** dialog, perform these steps to create the action—to show the **ThoughtErrorMessage**:

1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Show**.

2. Under **Configure actions**, under **Select the widget to hide/show** in the text field, type `Thought`.
3. Click on the checkbox next to **ThoughtErrorMessage**.
4. Click on **OK**.

We must finally hide the **ThoughtErrorMessage**, if the user interacts with the **EmailTextField**, **PasswordTextField**, or **NewUserLink**. Once we have defined all of the cases for the **Topic_Detail_Page_Header**, the **Widget Interactions and Notes** pane will look as follows:



Using the preceding screenshot as a reference, click on the header and perform the following steps:

1. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click on **OnMouseOutEmailTextField**. A **Case Editor** dialog box will open. In the **Case Name** field, type `ClearError`.
2. In the **Case Editor** dialog, to create the action to hide the **ThoughtErrorMessage**, perform the following steps:

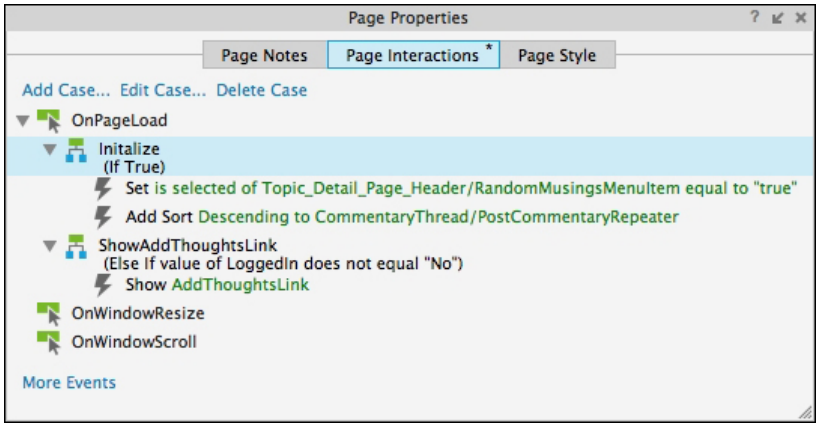
    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Hide**.
    2. Under **Configure actions**, under **Select the widget to hide/show** in the text field, type `Thought`.
    3. Click on the checkbox next to **ThoughtErrorMessage**.
    4. Click on **OK**.

3. Repeat steps 1 and 2, using the values given in the following table:

Event (Step 1)
OnMouseOutPasswordTextField
OnClickNewUser

We also need to sort the **PostCommentaryRepeater** by date in descending order. In the **Page Properties** pane under the design area, click on the **Page Interactions** tab. Under the **OnPageLoad** event, double-click on the **Initialize** case and perform the following steps to define the additional action to add **Add Sort**:

The following screenshot shows the **Case Editor** window with all the actions defined:



1. Under **Click to add actions**, scroll to the **Repeaters** dropdown, click on **Add Sort**.
2. Under **Configure actions**, under **Select the repeater to add sorting** click the checkbox next to **PostCommentaryRepeater**.
3. In the **Name** field, type `Descending`.
4. In the **Property** dropdown, select **CommentDate**.
5. In the **Sort as** dropdown, select **Date – MM/DD/YYYY**.
6. In the **Order** dropdown, select **Descending**.
7. Click on **OK**.

With commentary creation enabled for registered users, we will now allow new post creation for authorized users.

Allowing new post creation to authorized users

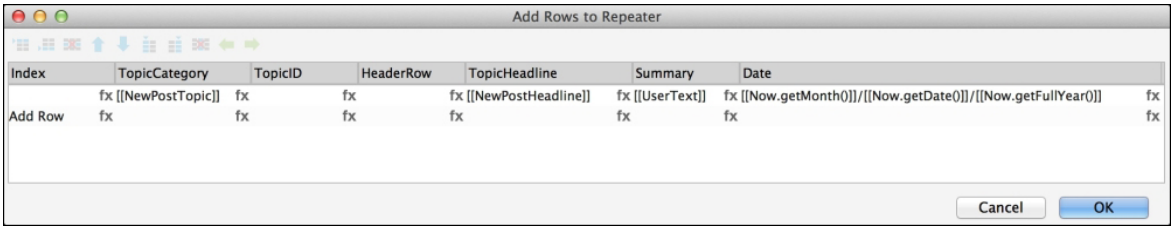
We will now enable new post creation on the **Random Musings** and **Topic List** pages for our authorized users. To allow commentary creation on the **Random Musings** page, we will need to place an **Add Post** link, as well as the **NewPost** master. When the authorized user clicks on the link, the **NewPost** master will be shown. The authorized user can then enter

their comment. When the authorized user clicks on the **Submit** button, the comment is added to the repeater.

To update the **Random Musings** page, in the **Sitemap** pane, double-click on the Page icon next to the **Random Musings** page to open in the design area. Perform the following steps:

1. The **ForumRepeater** currently in the design area does not reflect the changes we made to the **Forum** master. We must first delete the current **ForumRepeater** and place the updated **Forum** master. Click the **ForumRepeater** at coordinates (20,530) and press *Delete*.
2. In the **Masters** pane, drag the **Forum** master and drop it at coordinates (20,540) in the design area. In the **Widget Interactions and Notes** pane, click on the **Forum Name** field and type `PostRepeater`.
3. In the **Masters** pane, drag the **NewPost** master and drop it anywhere in the design area.
4. Go to the **Widget Interactions and Notes** pane, click on the **NewPost Name** field and type `NewPost`.
5. To create the **Add Row** action, double-click **OnClickSubmit**. In the **Case Editor** dialog, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Datasets** drop-down menu and click on **Add Rows**.
    2. Under **Configure actions**, click on the checkbox next to **ForumRepeater**.
    3. Click on the **Add Rows** button. In the **Add Rows to Repeater** dialog box, update row values for all columns as seen in the following screenshot:



6. To create the second action, to hide the **NewPostLightBoxDP**, perform the these steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Hide**.
    2. Under **Configure actions**, under **Select the widget to hide/show** in the text field, type `NewPost`.
    3. Click on **OK**.

7. In the **Widgets** pane, drag the **Label** widget and place it at coordinates (20,518). Type `Add new post`. In the toolbar, click on the checkbox next to **Hidden**. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `AddPostLabel`.
8. In the **Page Properties** pane under the design area, click on the **Page Interactions** tab. Click on **Add Case…**. In the **Case Name** field, type `AuthorizedUser`.
9. Create the condition. Click on the **Add Condition** button. In the **Condition Builder** dialog box, in the outlined condition box, perform these steps:

1. In the first dropdown, select **value of variable**.
2. In the second dropdown, select **LoggedIn**.
3. In the third dropdown, select **equals**.
4. In the fourth dropdown, select **value**.
5. In the text field, type `songwriter@test.com`.
6. Click on **OK**.

10. Create the action. To show the **AddPostLabel**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Show**.
    2. Under **Configure actions**, under **Select the widget to hide/show** in the text field, type `Add`.
    3. Under **NewComment**, click on the checkbox next to **AddThoughtsLink**.
    4. Click on **OK**.

11. Click on **AddPostLabel** at coordinates (20,518). In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `AddPostLabel`. In the **Widget Interactions and Notes** pane with the **Interactions** tab selected, click on **Add Case…**. A **Case Editor** dialog box will open.
12. Create the first action. To set panel state for the **NewPostLightBoxDP** dynamic panel, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Dynamic Panels** drop-down menu and click on **Set Panel State**.
    2. Under **Configure actions**, click on the checkbox next to **NewPostLightBoxDP**.
    3. Next to **Select the state**, click on the dropdown and select **State2**.

13. Create the second action; to show the **NewPostLightBoxDP**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown, click on the **Show/Hide** dropdown and click on **Show**.
    2. Under **Configure actions**, click on the checkbox next to **NewPostLightBoxDP**.
    3. Click on **OK**.

14. To update the **Topic List** page, in the **Sitemap** pane, double-click on the Page icon next to the **Topic List** page to open in the design area. Repeat the preceding steps 1-13 to complete the updates for the **Topic List** page.

Note

Once you have completed the preceding steps, move the **AddPostLabel** to coordinates (20,120). Move the **PostRepeater** to coordinates (20,142).

Congratulations, we have completed all the pages of our blog site.

Summary

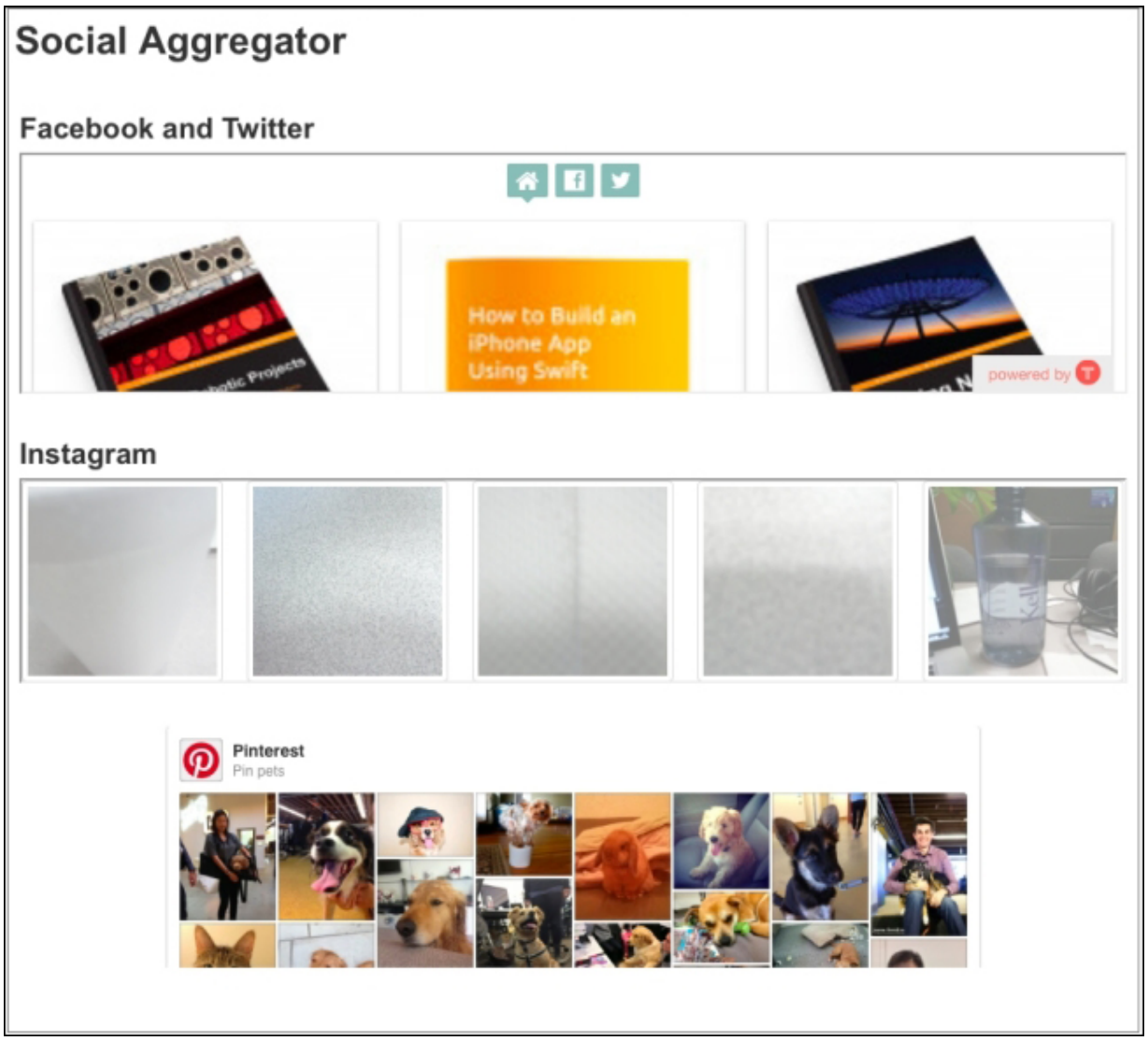
We learned how to leverage masters and pages from our community site to create a new blog site. We explored user account validation, dynamic account creation, and means to provide additional privileges to authorized users. We learned how to simulate light boxes, how to store user input into global variables, and populate data to added rows in repeater widgets.

In the next chapter, we will explore building a content aggregator.

Chapter 4. Leveraging Social Content Aggregation

Social media sites ebb and flow in popularity. As a result, companies and individuals are constantly evaluating their social strategies and determining which social campaign strategies and channels are most beneficial. In an effort to maximize investment in social media, the content from these social channels is often displayed on a website in the form of a social media hub. The process of gathering content from various sources across the web is often referred to as **aggregation**.

For this blueprint, we will explore leveraging Facebook, Twitter, Instagram, and Pinterest in an Axure RP 7 prototype. Once completed, our aggregator will look as follows:



In this chapter, we will learn about:

- Developing an aggregator
- Building our Social Content master
    - Adding Facebook and Twitter
    - Pulling Instagram content
    - Including a Pinterest board

Developing an aggregator

We will create a master for our **Social Content Aggregator**. This will provide us the opportunity to reuse and will act as a central location to make global updates on all pages using the master.

To build our aggregator, we will leverage two different approaches. The first approach will leverage the **Inline Frame** widget with external content aggregators (that is, **Tint** and **SnapWidget**). In the second approach, we will create a plugin for **AxShare**, allowing us to leverage external JavaScript that will allow us to pull external feeds into our prototypes.

> ## Note
>
> **AxShare** is Axure's free prototype hosting service. Learn more at http://axshare.com.

Building our Social Content master

In Chapter 2, *Building a Community Site*, we used a social aggregator, **Tint**, with an **Inline Frame** widget to provide Facebook and Twitter feeds to a Social Media widget. For this blueprint, in addition to **Tint**, we will leverage **SnapWidget** for Instagram. Next, we will use the plugin feature of AxShare to pull in a Pinterest board.

To create our **Social Content** master, create a new Axure RP document and perform the following steps:

1. In the **Masters** pane, click on the Add Master icon, type `Aggregator` and press *Enter*.
2. In the **Masters** pane, double-click on the icon next to the **Aggregator** master to open in the design area.

We are now ready to add the Facebook and Twitter feeds.

Adding Facebook and Twitter to our feed

We will use an **Inline Frame** widget with the **Tint** social aggregator (http://www.tintup.com) to add Facebook and Twitter feeds.

Note

After we have created a Tint account and added a Tint (that is, social media feed), a URL will be provided for our feed. You will need this URL to complete the following steps. You can get your URL from the Chapter 2, *Building a Community Site*, prototype file or by logging in to your Tint account.

To add the Facebook and Twitter feed, perform the following steps:

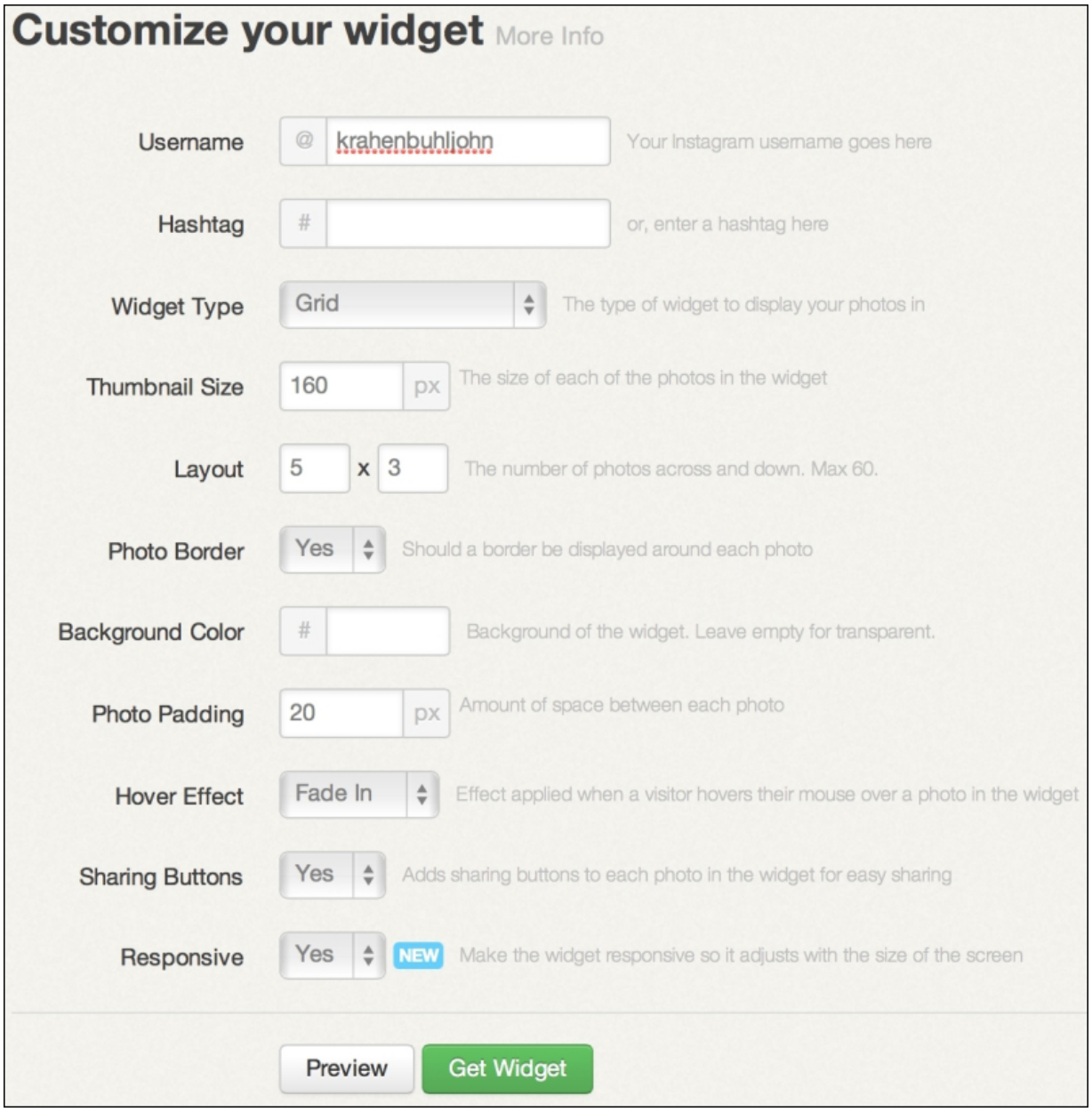
1. From the **Widgets** pane, drag the **Rectangle** widget  and place at coordinates (10,0). In the toolbar, change **w:** to `940` and **h:** to `850`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `SocialMediaFeedBackground`.
2. From the **Widgets** pane, drag the **Heading 1** widget  and place at coordinates (16,9). Type `Social Aggregator`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Headline`.
3. From the **Widgets** pane, drag the **Heading 2** widget  and place at coordinates (20,86). Type `Facebook and Twitter`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `SocialMediaFeedHeading1`.
4. From the **Widgets** pane, drag the **Inline Frame** widget  and place at coordinates (20,120). Perform the following steps:
    1. In the **Widget Interactions and Notes** pane, click on the **Inline Frame Name** field and type `SocialMediaFeedIF1`.
    2. In the toolbar, change **w:** to `920` and **h:** to `200`.
    3. Right-click on the **SocialMediaFeedIF1 Inline Frame** widget, click on **Scrollbars**, and click on **Never Show Scrollbars**.

4. Right-click on the **SocialMediaFeedIF1 Inline Frame** widget, click on **Frame Target**. In the **Link Properties** dialog box, click the radio button next to **Link to an external url or file**. In the **Hyperlink** field, paste the embed link to your Tint (for example. http://www.tintup.com/axuredemo).

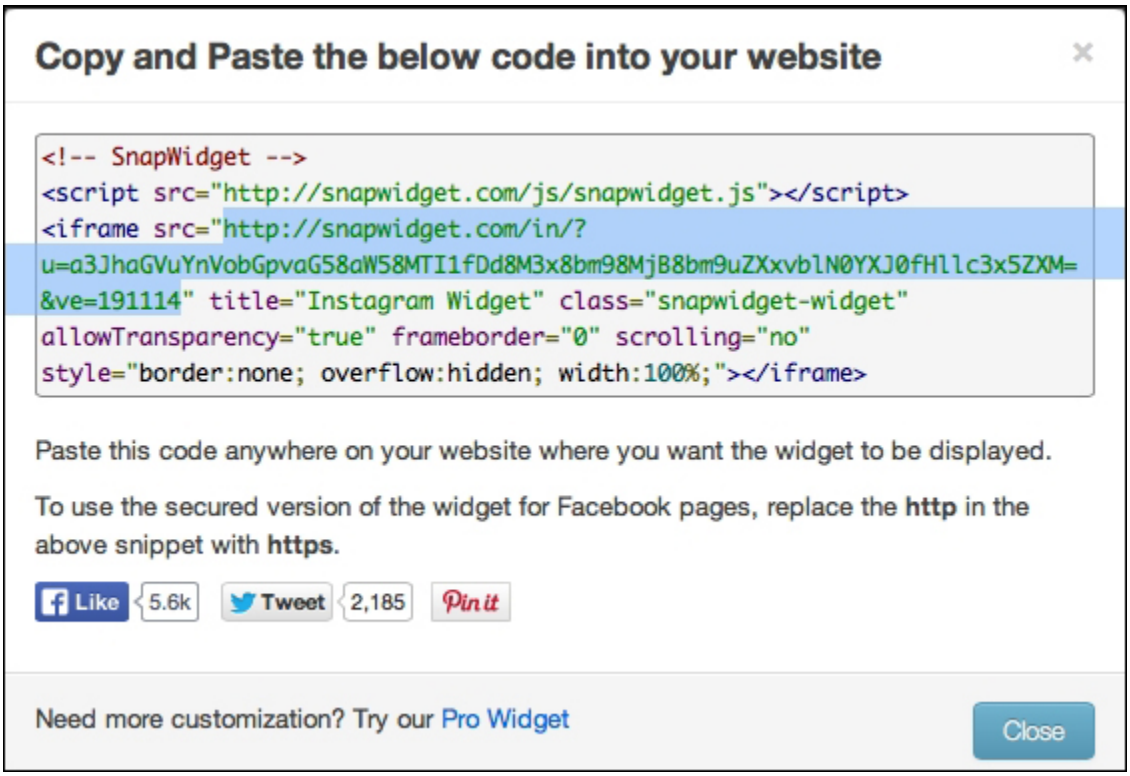
With the Facebook and Twitter feed completed, we are now ready to add the Instagram feed.

Pulling Instagram content

To display Instagram content on our aggregator, we will use an **Inline Frame** widget with the **SnapWidget** Instagram widget (http://snapwidget.com). To customize your own **SnapWidget**, click the **Get Your Free Widget** button on the **SnapWidget** home page and complete the **Customize your widget** form. The following is an example screenshot of the completed form:



Click the **Get Widget** button and copy the URL shown in the pop-up window. The URL will typically be shown on the third line starting with `src=`. The following is an example screenshot highlighting the URL:



Note

In the **Customize your widget** form, make sure to clear the **Hashtag** field to pull all Instagram content associated with the Instagram username.

Since we have our **SnapWidget** URL, we are now ready to include an Instagram feed into our **Social Media** master. Perform the following steps:

1. From the **Widgets** pane, drag the **Heading 2** widget H2 and place it at coordinates (20,356). Type `Instagram`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `SocialMediaFeedHeading2`.
2. From the **Widgets** pane, drag the **Inline Frame** widget and place at coordinates (20,390). Perform these steps:

1. In the **Widget Interactions and Notes** pane, click on the **Inline Frame Name** field and type `SocialMediaFeedIF2`.
2. In the toolbar change, **w:** to `920` and **h:** to `200`.
3. Right-click on the **SocialMediaFeedIF2 Inline Frame** widget, click on **Scrollbars**, and click on **Never Show Scrollbars**.
4. Right-click on the **SocialMediaFeedIF2 Inline Frame** widget, click on **Frame Target**. In the **Link Properties** dialog box, click the radio button next to **Link to an external url or file**. In the **Hyperlink** field, paste the embed link to your **SnapWidget**.

For example, http://snapwidget.com/in/?u=a3JhaGVuYnVobGpvaG58aW58MTYwfDV8M3x8eWVzfDIwfGZhZGVJbnxvbIN0YXJ0fHllc3x5ZXM=&ve=201114.

With the Instagram feed integrated, we are now ready to include a Pinterest board.

Including a Pinterest board

To display Instagram content within our aggregator, we will use a dynamic panel widget and Pinterest's widget builder. Pinterest's widget builder will generate custom JavaScript. To include this in our prototype, we will create a plugin for AxShare that will include Pinterest's reference link and JavaScript inside the **PinterestDP** dynamic panel element.

To add a **PinterestDP** dynamic panel element to our **Social Media** master and publish to AxShare, perform the following steps:

1. From the **Widgets** pane, drag the **Dynamic Panel** widget and place it at coordinates (20,596). In the toolbar, change **w:** to `920` and **h:** to `200`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `PinterestDP`.
2. In the **Sitemap** pane, double-click on the Page icon next to the **Home** page to open in the design area.

3. From the **Masters** pane, drag the **Aggregator** master and place at coordinates (0,0).
4. In the main toolbar, navigate to **Publish** | **Publish to AxShare**. If you have an existing account, enter your e-mail and password. If you do not have an account, click the **Create Account** tab and enter your information. Click on the **Create a new project** radio button and in the **Name** field, enter the name for the prototype (that is, `Chapter04`).

To generate the Pinterest's reference link and JavaScript, go to http://business.pinterest.com/widget-builder/ and click on the **Board Widget**. Complete the **Board Widget** form and click on the **Build It** button. Copy all of the code in the **Copy the code** section. The following screenshot shows a completed form with the code section:

Pin It Button

Pindiana Jones

Follow Button

Pin Widget

Profile Widget

Board Widget

Board widget: Show up to 30 of your favorite board's latest Pins.

Pinterest Board URL:

http://www.pinterest.com/pinterest/pin-pets/

Image Width:

80

Board Height:

170

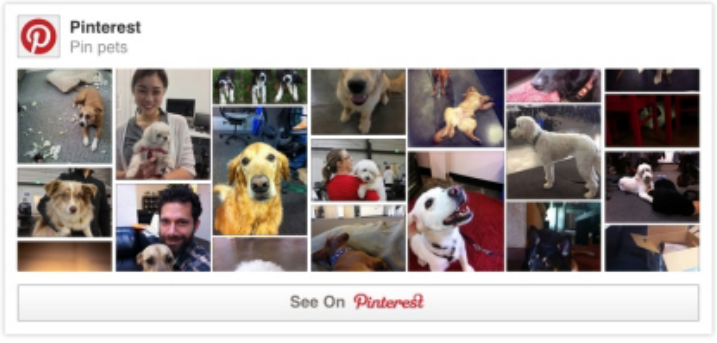
Board Width:

920

Custom Sizes: Square Sidebar Header Roll Your Own

Build It!

Preview

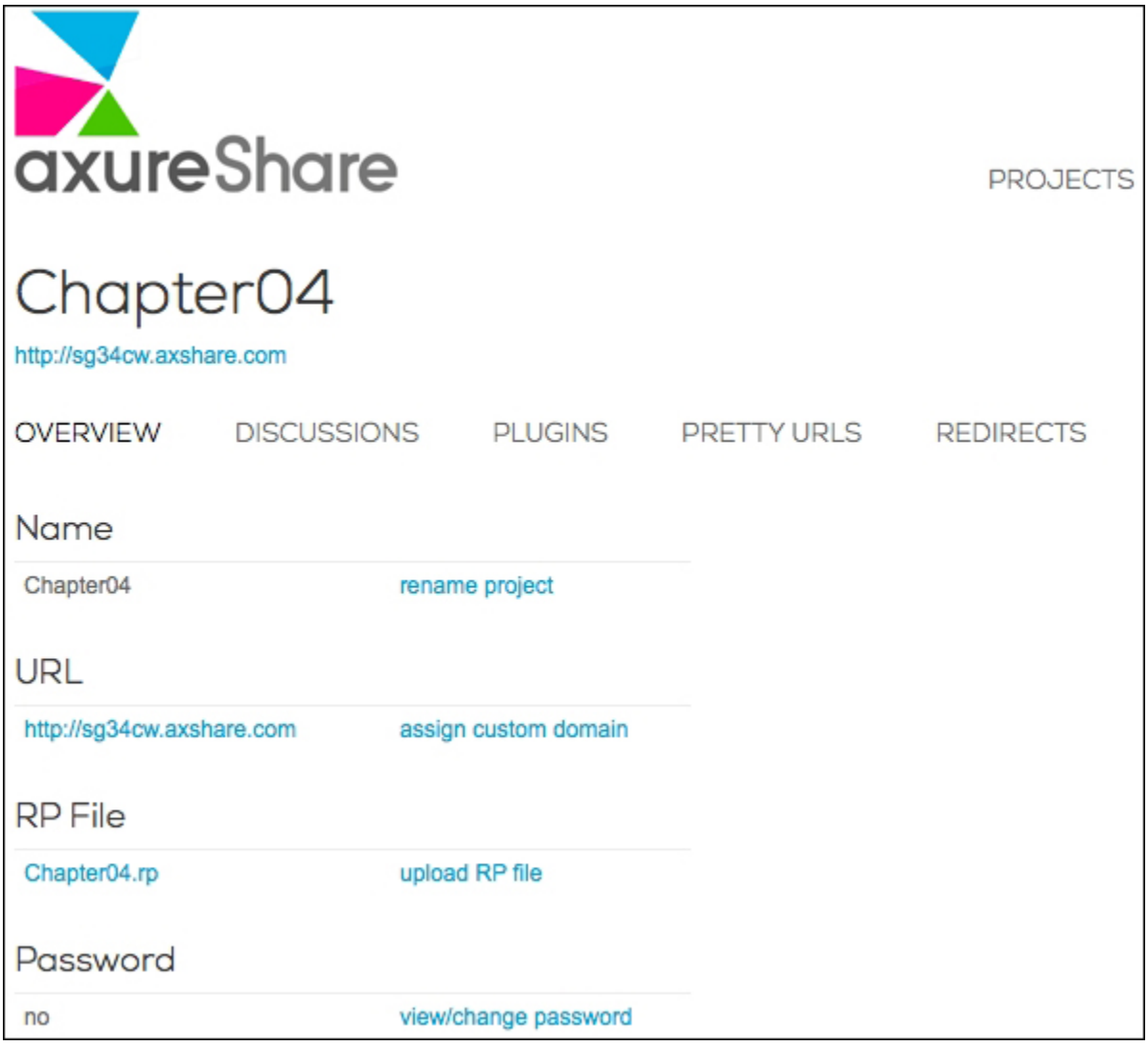


Copy the code

Copy and paste this code into your page where you want your button or widget to appear:

```
<a data-pin-do="embedBoard" href="http://www.pinterest.com/pinterest/pin-pets/"
data-pin-scale-width="80" data-pin-scale-height="170" data-pin-board-
width="920">Follow Pinterest's board Pin pets on Pinterest.</a>
<!-- Please call pinit.js only once per page -->
<script type="text/javascript" async src="//assets.pinterest.com/js/pinit.js">
</script>
```

We now are ready to create our Pinterest plugin for AxShare. The plugin for AxShare feature allows us to place snippets of code inside a specific dynamic panel. Our plugin will include the Pinterest reference link with JavaScript and will be placed inside the **PinterestDP** dynamic panel element. Log in to your AxShare account at http://axshare.com. Click on the name of your prototype to edit the project settings. Your project settings will appear similar to the following screenshot:



Click on the **PLUGINS** menu item. Perform these steps:

1. Click on the **New Plugin** button.
2. In the **Add New Plugin** dialog, perform the following:

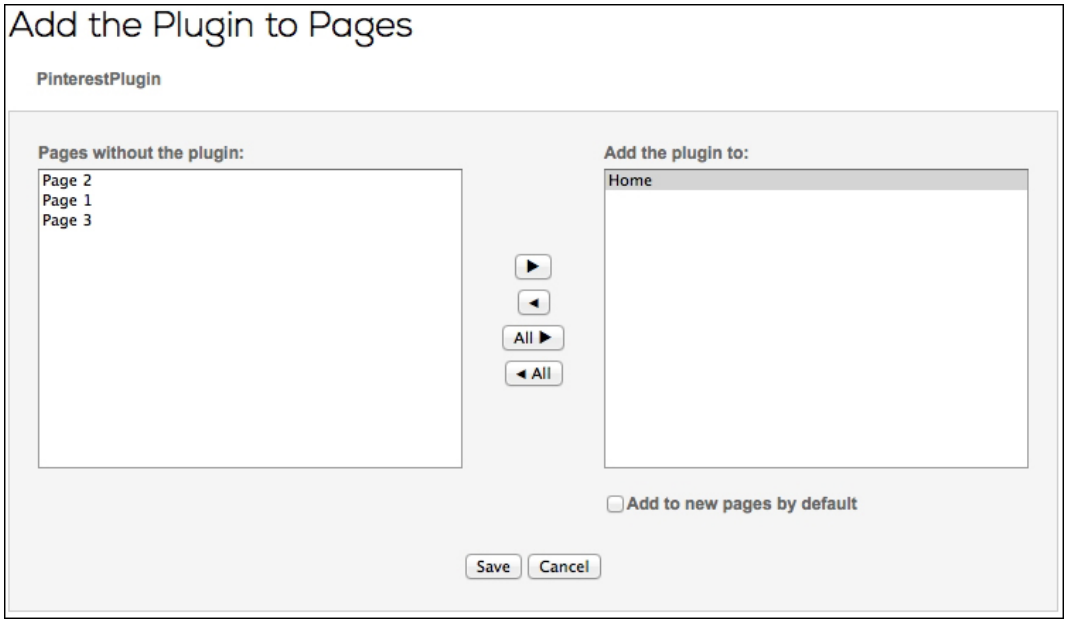
1. In the **Plugin Name** field, enter `PinterestPlugin`.
2. Under **Location:**, click on the radio button next to **Inside Dynamic Panels with Name:** and in the text field, enter `PinterestDP`.
3. In the **Content:** text area, paste the Pinterest's reference link and JavaScript. For example:

```
<a data-pin-do="embedBoard"
href="http://www.pinterest.com/pinterest/
pin-pets/" data-pin-scale-width="80"
data-pin-scale-height="170" data-pin-
board-width="920">Follow Pinterest's
board Pin pets on Pinterest.</a>
<!-- Please call pinit.js only once per
page -->
<script type="text/javascript" async
src="//assets.pinterest.com/js/pinit.js">
</script>
```

4. Click on the **Save and continue** button.

3. In the **Add the Plugin to Pages** dialog, do these steps:

1. In the **Pages without the Plugin:** text area, click on **Home**.
2. Click on the Add Page button ▶. You will see the **Home** page link appear under **Add the plugin to:**. Your **Add the Plugin to Pages** dialog will look similar to this:



3. Click on the **Save** button.
4. From the **Project Settings** page, under the **Project name** click on the link to view your prototype.

With our sample Pinterest board functional, we have completed our **Aggregator** master for social content channels.

Summary

We created a **Social Content Aggregator** master that included feeds from Facebook, Twitter, Instagram, and Pinterest. To accomplish this, we used **Inline Frame** widgets, as well as created a Pinterest plugin for AxShare. Content from Facebook, Twitter, and Instagram were supplied via integration to tools such as **Tint** and **SnapWidget**. Pinterest content was pulled directly from Pinterest using the open Pinterest API.

In the next chapter, we will explore building a sample Portfolio page.

Chapter 5. Showcasing Portfolio Content

With the rise of efficient content delivery networks, we continue to see more immersive online experiences. Individuals leverage full width images that showcase their individuality and work experiences. One popular interactive paradigm is the use of **parallax scrolling** in web design. A parallax site will have a background image scroll at a differing rate than content in the foreground. For this chapter, we will build a parallax portfolio site that will look as follows:



In this chapter, we will learn about:

- Designing our parallax site
- Placing page anchors
- Building the background dynamic panel
- Creating our sectional content
- Adding additional portfolio interactions
- Updating page interactions

Designing our parallax site

We will leverage a minimalist design keeping the viewer's focus on a brief synopsis of our work history and compelling the viewer to

contact us via e-mail. Our site will have the following three sections, as reflected in the left navigation:

- **Work**
- **Contact**
- **About**

To enable our parallax effect, we will have foreground content organized per section in front of a background dynamic panel called **BackgroundDP**. The **BackgroundDP** will contain an individual dynamic panel per section. Each individual dynamic panel will have a back image set to **Stretch to Cover**. The **Stretch to Cover** setting will stretch the back image to cover the entire width and height of the dynamic panel.

> ## Note
>
> Learn more about options to change the page style at https://www.axure.com/learn/basic/page-style.

When the user scrolls past the given vertical scroll points, the left navigation will update to represent the section that is currently being displayed. The user may also choose to click on a given left navigation menu item. The term page anchor (or anchor link) refers to a widget that is used as an *anchor* for a specific area of the page. When a menu item is clicked, the page will scroll to the corresponding page anchor.

We will start our design by first placing our page anchors. Next, we will place the **BackgoundDP** dynamic panel and the corresponding section dynamic panels. Finally, we will place each section's content.

Placing page anchors

We will use three Hot Spot widgets as page anchors. When the user clicks a corresponding menu item, the page will move to the appropriate page anchor. First, open a new Axure RP 7 project with the home page opened in the design area. To place each section's page anchors, perform the following steps:

1. From the **Widgets** pane, drag and drop the **Hot Spot** widget  and place it at the coordinates (860,0). In the **Widget Interactions and Notes** pane, click on the **Hot Spot Name** field and type `MyWorkAnchor`.
2. Drag and drop the **Hot Spot** widget  and place it at the coordinates (860,500). In the **Widget Interactions and Notes** pane, click on the **Hot Spot Name** field and type `ContactAnchor`.
3. Drag and drop the **Hot Spot** widget  and place it at the coordinates (860,1690). In the **Widget Interactions and Notes** pane, click on the **Hot Spot Name** field and type `AboutMeAnchor`.

> ## Note
>
> The location of page anchors can be optimized depending on your target browser's vertical height.

We are now ready to build **BackgroundDP**.

Building the background dynamic panel

The **BackgroundDP** dynamic panel contains three dynamic panels, one for each section. Each individual dynamic panel contains a back image that is set to **Stretch to Cover**. To place the nested dynamic panels, perform these steps:

1. Drag and drop the **Dynamic Panel** widget and place it at the coordinates (80,0). In the toolbar, change **w:** to `830`, **h:** to `2490`. In the **Widget Interactions and Notes** pane, click in the **Dynamic Panel Name** field and type `BackgroundDP`. In the **Widget Manager** pane, under the **BackgroundDP**, slow-click the text beside the State icon to rename and type `BackgroundImages`.
2. From the **Widget Manager** pane, under **BackgroundDP**, click on the State icon beside **BackgroundImages** twice to open in the design area.
3. Perform the following steps:

    1. Drag and drop the **Dynamic Panel** widget and place it at the coordinates (0,0). In the toolbar, change **w:** to `830`, **h:** to `830`. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `WorkBackImageDP`.
    2. Under the design area with the **Panel State Style** tab selected next to **Back Image**, click on the **Import** button and select your image.
    3. Next to **Repeat**, click on **Stretch to Cover** from the dropdown.

4. Repeat step 3 two more times, using the values given in the following table:

Coordinates	**Name Field**
(0,830)	**ContactBackImageDP**
(0,1660)	**AboutBackImageDP**

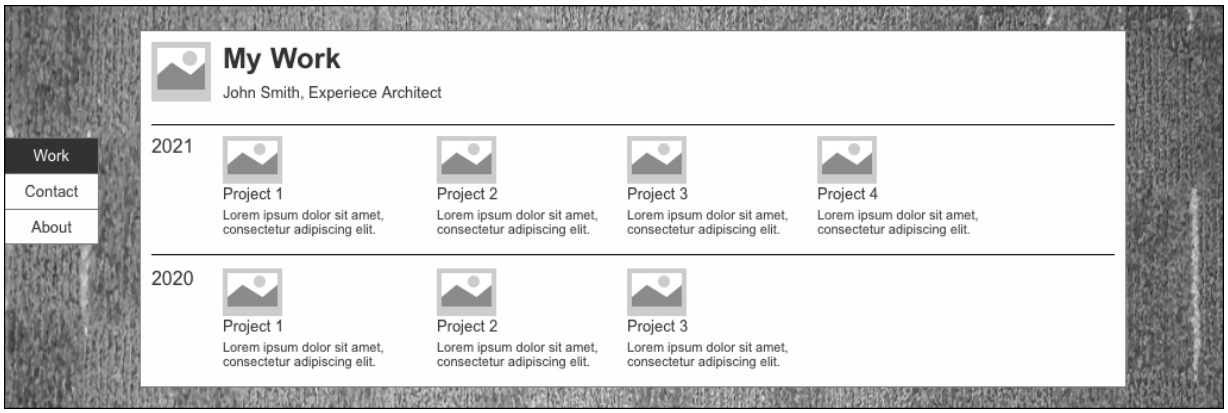
Now, we will create each section's content.

Creating our sectional content

Our site will have three sections as follows:

- **Work**
- **Contact**
- **About**

When completed, our **Work** section will look as follows:



We will first create an **ExampleProject** master. To do this, perform the following steps:

1. In the **Masters** pane, click on the Add Master icon, type `ExampleProject` and press *Enter*.
2. Again, in the **Masters** pane, double click on the icon next to the **ExampleProject** master to open in the design area.
3. Drag and drop the **Image** widget and place it at the coordinates (0,0). In the toolbar, change **h:** to `40`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ProjectImg`.
4. Drag and drop a **Label** widget onto the design area and place it at the coordinates (0,41). Type `Project`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ProjectLabel`.

5. Drag and drop a **Label** widget onto the design area and place it at the coordinates (0, 60). Type `Lorem ipsum dolor sit amet, consectetur adipiscing elit.`. In the toolbar, change **w:** to `150` and **h:** to `24`. In the **Widget Interactions and Notes** pane, click in the **Shape Name** field and type `ProjectCopy`. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Font** and change the font size to `11`.

With our **ExampleProject** master completed, we are now ready to create the **Work** section. In the **Sitemap** pane, double-click on the Page icon next to the **Home** page to open in the design area. To create the **Work** section, perform the following steps:

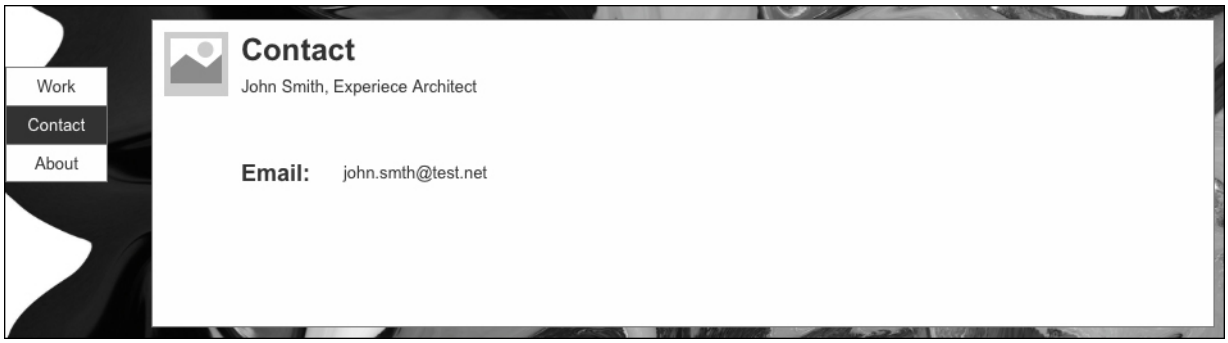
1. Drag and drop the **Rectangle** widget and place it at the coordinates (80, 200). In the toolbar, change **w:** `830` and **h:** `300`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `MyWorkRectangle`.
2. Drag and drop the **Image** widget and place at the coordinates (90,210). In the **Widget Interactions and Notes** pane, click in the **Shape Name** field and type `MyWorkImg`.
3. Drag and drop the **Heading 2** widget H2 and place at the coordinates (150,210). Type `My Work`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `MyWorkHeadline`. In the **Widget Properties and Style** pane, with the **Style** tab selected, scroll to **Font** and change the font size to `24`.
4. Drag and drop a **Label** widget onto the **Design Area** and place at the coordinates (150,245). Type `John Smith, Experience Architect`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `NameTitleLabel1`.
5. Drag and drop the **Horizontal Line** widget and place at the coordinates (90,274). In the toolbar, change **w:** to `810`.
6. Drag and drop the **Horizontal Line** widget and place at the coordinates (90,383). In the toolbar, change **w:** to `810`.
7. Drag and drop a **Label** widget onto the design area and place at the coordinates (90,289). Type `2021`. In the **Widget**

Interactions and Notes pane, click on the **Shape Name** field and type `YearMostRecent`.

8. Drag and drop a **Label** widget onto the design area and place at the coordinates (90,400). Type `2020`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `YearPrior`.
9. From the **Masters** pane, drag and drop the **ExampleProject** master and place at the coordinates (150,289).
10. Repeat step 9 six more times, using the coordinates as outlined in the following table:

Coordinates
(330,289)
(490,289)
(650,289)
(150,400)
(330,400)
(490,400)

With the **Work** section completed, we are now ready to build the **Contact** section with interactive e-mail link. When completed, our **Contact** section will look as follows:



To create the **Contact** section, perform these steps:

1. Drag and drop the **Rectangle** widget and place it at the coordinates (80,750). In the toolbar, change **w:** to `830` and **h:** to `240`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ContactRectangle`.
2. Drag and drop the **Image** widget and place it at the coordinates (90,760). In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ContactImg`.
3. Drag and drop the **Heading 2** widget H2 and place it at the coordinates (150,760). Type `Contact`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ContactHeadline`. In the **Widget Properties and Style** pane, with the **Style** tab selected scroll to **Font** and change the font size to `24`.
4. Drag and drop a **Label** widget onto the design area and place at the coordinates (150,795). Type `John Smith, Experience Architect`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `NameTitleLabel2`.
5. Drag and drop a **Label** widget onto the design area and place at the coordinates (150,860). Type `Email:`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `eMailLabel`. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Font** and change the font size to `18`.
6. Drag and drop a **Label** widget onto the design area and place at the coordinates (230,862). Type `john.smith@test.net`.

In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `MailToLink`.

7. To define the **MailToLink OnClick** interaction, in the **Widget Interactions and Notes** pane with the **Interactions** tab selected, double-click on **OnClick**. In the **Case Editor** dialog, perform the following steps:

    1. Under **Click to add actions**, navigate to **Links** | **Open Link** and click on **Current Window**.
    2. Under **Configure actions**, click on the radio button next to **Link to an external url or file**. Beside the **Hyperlink** field, click on the **fx** icon to open the **Edit Value** dialog box.
    3. Under **Insert Variable or Function…**, type `mailto:[[LVAR1]]`.
    4. Beneath **Local Variables** under **Add Local Variable**, in the text field type `LVAR1`.
    5. In the first dropdown, select **text on widget**.
    6. In the second dropdown, select **This**.
    7. Click on **OK**.
    8. Click on **OK**.

With the **Contact** section completed, we are now ready to build the **About** section with interactive e-mail link. When completed, our **About** section will look as follows:



To create the **About** section, perform the following steps:

1. Drag and drop the **Rectangle** widget and place at the coordinates (80,1310). In the toolbar, change **w:** to `830` and **h:** to

`240`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `AboutMeRectangle`.

2. Drag and drop the **Image** widget  and place at the coordinates (90, 1320). In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `AboutMeImg`.
3. Drag and drop the **Heading 2** widget  and place at the coordinates (150,1320). Type `Amount Me`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `AboutMeHeadline`. In the **Widget Properties and Style** pane, with the **Style** tab selected scroll to **Font** and change the font size to `24`.
4. Drag and drop a **Label** widget  onto the design area and place at the coordinates (150, 1355). Type `John Smith, Experience Architect`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `NameTitleLabel3`.
5. Drag and drop the **Paragraph** widget  and place at the coordinates (150,1400). In the toolbar, change **w:** to `720` and **h:** to `60`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `AboutMeCopy`.

With each section's content designed, we are now ready to place the left navigation and add additional interactions.

Adding additional portfolio interactions

By default, the first menu item is shown in the **Selected** interaction style. When the viewer hovers the cursor over a menu item, this is displayed in the **MouseOver** interaction style. If the viewer clicks the menu item, it is transitioned to the **Selected** interaction style, the page advances to the associated page anchor and all other menu items are placed in the unselected state.

First, we will place the left navigation and define interactions for each menu item. We will then define the remaining page interactions. To place the left navigation and define interactions, perform these steps:

1. From the **Widgets** pane, under the **Menus and Table** dropdown, drag the **Classic Menu - Vertical** widget  and place at the coordinates (0,0). In the toolbar, change **w:** to `80` and **h:** to `90`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `LeftNav`. Right-click on the gray outline around the **LeftNav** at the coordinates (0,0) and click on **Interaction Styles…**.

   > ## Note
   >
   > To bring up the menu for the **LeftNav** widget, move the mouse over the gray outline until you see the cursor turn into a four-way arrowhead. Once you see the four-way arrowhead, right-click.

2. In the **Set Interaction Styles** dialog, perform the following:

   1. Click on the **MouseOver** tab. Click on the checkbox next to Fill Color. Click on the down arrow next to the Fill Color icon

. In the drop-down menu, in the **#** text field, enter `F2F2F2`. Click off of the dropdown to close it.

2. Now, click on the **Selected** tab.
3. Click on the checkbox next to Fill Color. Click on the down arrow next to the Font Color icon . In the drop-down menu, in the **#** text field, enter `FFFFFF`. Click off of the dropdown to close it.
4. Click on the checkbox next to Fill Color. Click on the down arrow next to the Fill Color icon . In the drop-down menu, in the **#** text field, enter `333333`. Click off of the dropdown to close it.
5. Click on **OK**.

3. Right-click on the gray outline around the **LeftNav** at the coordinates (0,0) and click on **Convert to Dynamic Panel**.
4. In the **Widget Properties and Style** pane, click on the **Properties** tab. Click on the **Pin to Browser** link. Click on the checkbox next to **Pin to browser window**. Under **Horizontal Pin**, click on the radio button next to **Left**. Under **Vertical Pin**, click on the radio button next to **Middle**. Click on the checkbox next to **Keep in front (browser only)**. Click on **OK**.
5. In the **Widget Manager** pane, slow-click the text beside the unnamed dynamic panel icon  to rename and type `LeftNavDP`.
6. In the **Widget Manager** pane, under the **LeftNavDP** dynamic panel, click on the State icon  beside **State1** twice to open in the design area.
7. Click on the first menu item and type `Work`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `WorkMenuItem`. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click on **OnClick**. A **Case Editor** dialog box will open.
8. Now, let's create the first action. To define the **Scroll to Widget** action, perform the following steps:

    1. Under **Click to add actions**, go to the **Links** drop-down menu and click on **Scroll to Widget (Anchor Link)**.

2. Under **Configure actions**, click on the checkbox next to **MyWorkAnchor**.

9. Create the second action. To set **Selected** to **True** for the **WorkMenuItem**, perform these steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **WorkMenuItem**.
    3. Under **Set selected state to**, verify the first dropdown is set to **value** and the second dropdown is set to **true**.

10. Create the third action. To set **Selected** to **False** for the **ContactMenuItem** and **AboutMenuItem**, perform the following:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **ContactMenuItem**.
    3. Under **Set selected state to**, verify that the first dropdown is set to **value** and the second dropdown is set to **false**.
    4. Under **Configure actions**, click on the checkbox next to **AboutMenuItem**.
    5. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
    6. Click on **OK**.

11. Click the second menu item and type `Contact`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `ContactMenuItem`. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click on **OnClick**. A **Case Editor** dialog box will open.
12. Create the first action for this menu item. To define the **Scroll to Widget** action, do the following steps:

1. Under **Click to add actions**, go to the **Links** drop-down menu and click on **Scroll to Widget (Anchor Link)**.
2. Under **Configure actions**, click on the checkbox next to **ContactAnchor**.

13. Create the second action. To set **Selected** to **True** for the **ContactMenuItem**, perform these steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **ContactMenuItem**.
    3. Under **Set selected state to**, verify the first dropdown is set to **value** and the second dropdown is set to **true**.

14. Create the third action. To set **Selected** to **False** for the **WorkMenuItem** and **AboutMenuItem**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **WorkMenuItem**.
    3. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
    4. Under **Configure actions**, click the checkbox next to **AboutMenuItem**.
    5. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
    6. Click on **OK**.

15. Click on the third menu item and type `About`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `AboutMenuItem`. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click on **OnClick**. A **Case Editor** dialog box will open.

16. Create the first action. To define the **Scroll to Widget** action, perform the following steps:

    1. Under **Click to add actions**, go to the **Links** drop-down menu and click on **Scroll to Widget (Anchor Link)**.
    2. Under **Configure actions**, click on the checkbox next to **AboutAnchor**.

17. Create the second action. To set **Selected** to **True** for the **AboutMenuItem**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked**, drop-down menu click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **AboutMenuItem**.
    3. Under **Set selected state to**, verify that the first drop-down is set to **value** and second drop-down is set to **true**.

18. Create the third action. To set **Selected** to **False** for the **WorkMenuItem** and **ContactMenuItem**, do as follows:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **WorkMenuItem**.
    3. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
    4. Under **Configure actions**, click the checkbox next to **ContactMenuItem**.
    5. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
    6. Click on **OK**.

Note

Axure has many ways to accomplish the same task.

In this example, we explored using a **Classic Menu – Vertical** widget for our **LeftNav**. For the **OnClick** event of each menu item, we need to **set the selected state** of the clicked menu item to **true** and then **set the selected state** of each additional menu item to **false** to create the desired effect. This is a tedious manual approach, if you have numerous menu items.

One alternative approach for this example is to use **Rectangle widgets** to create **LeftNav**. Then, in the **Widget Properties and Style** pane with the **Properties** tab selected, create a **selection group**. In future, you could assign any new Rectangle widgets to the **selection group** and Axure will manage the selected states for you.

Learn more about this topic at http://www.axure.com/learn/dynamic-panels/basic/tab-control-tutorial.

With the additional portfolio interactions completed, we are now ready to define the remaining page interactions.

Updating page interactions

Our portfolio page will have interactions defined for two events: **OnPageLoad** and **OnWindowScroll**. The **OnPageLoad** event will set the default menu item to its selected state (that is, **WorkMenuItem** selected). The **OnWindowScroll** event will control the parallax scrolling effect of the **BackgroundDP** and update the selected menu item of **LeftNavDP**. We will first define the case for the **OnPageLoad** event. Next, we will define the four cases for the **OnWindowScroll** event.

Defining the OnPageLoad event

In the **Sitemap** pane, double-click on the Page icon next to the **Home** page to open in the design area. To enable actions for the **OnPageLoad** event, in the **Page Properties** pane under the design area, click on the **Page Interactions** tab. Click on **Add Case…**.A **Case Editor** dialog box will open. In the **Case description** field, type `Set Selected Default Left Nav Item`.

Now, let's create the action. To set **Selected** to **True** for the **WorkMenuItem**, perform the following steps:

1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
2. Under **Configure actions**, click on the checkbox next to **WorkMenuItem**.
3. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **true**.
4. Click on **OK**.

With the **OnPageLoad** event defined, we are now ready to enable the **OnWindowScroll** interactions.

Enabling OnWindowScroll interactions

We will now define the **WindowScrolled** case for the **OnWindowScroll** event. The **WindowScrolled** case will control the parallax scrolling effect and move the **BackgroundDP**.

Defining the WindowScrolled case

In the **Page Properties** pane under the design area, click on the **Page Interactions** tab. Double-click the **OnWindowScroll** event. A **Case Editor** dialog box will open. In the **Case description** field, type `WindowScrolled`.

Create the action. To move the **BackgroundDP**, perform the following steps:

1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu, and click on **Move**.
2. Under **Configure actions**, click on the checkbox next to **BackgroundDP**.
3. Next to **Move**, click on the dropdown and click on **to**.
4. In the **y:** field, enter `-[[Window.scrollY / 1.5]]`.
5. Click on **OK**.

> ### Note
>
> The divisor controls the rate at which the **BackgroundDP** scrolls. Increase this constant to increase the scrolling effect.

With the **WindowScrolled** case defined, we are now ready to complete the remaining **OnWindowScroll** cases.

Completing the remaining OnWindowScroll cases

There are three remaining **OnWindowScroll** cases. The cases with brief descriptions are as follows:

OnWindowScroll case	**Description**
WindowScroll Y < 400	The **WorkMenuItem** selected state is set to **true**. The **ContactMenuItem** and **AboutMenuItems** selected state are set to **false**
WindowScroll Y > 400	The **ContactMenuItem** selected state is set to **true**. The **WorkMenuItem** and **AboutMenuItems** selected state are set to **false**.
WindowScroll Y > 800	The **AboutMenuItems** selected state is set to **true**. The **WorkMenuItem** and **ContactMenuItem** selected state are set to **false**.

We will now define the **WindowScroll Y < 400** case.

Defining the WindowScroll Y < 400 case

In the **Page Properties** pane, under the design area, click on the **Page Interactions** tab. Double-click on the **OnWindowScroll** event.

A **Case Editor** dialog box will open. In the **Case description** field, type `WindowScroll Y < 400`. In the **Case Editor** dialog, perform these steps:

1. Add the first condition: Click on the **Add Condition** button. In the **Condition Builder** dialog box, go to the outlined condition box and perform the following steps:

    1. In the first dropdown, select **value**.
    2. In the second dropdown, select **[[Window.scrollY]]**.
    3. In the third dropdown, select **is greater than**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth dropdown, select **0**.

2. To add the second condition, click on the green **+** sign. In the **Condition Builder** dialog box, go to the outlined condition box and perform these steps:

    1. In the first dropdown, select **value**.
    2. In the second dropdown, select **[[Window.scrollY]]**.
    3. In the third dropdown, select **is less than**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth dropdown, select **400**.
    6. Click on **OK**.

3. Create the first action. To set **Selected** to **True** for the **WorkMenuItem**, do the following:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **WorkMenuItem**.
    3. Under **Set selected state to**, verify that the first dropdown is set to **value** and the second dropdown is set to **true**.

4. Create the second action. To set **Selected** to **False** for the **ContactMenuItem** and **AboutMenuItem**, perform the following

steps:

1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
2. Under **Configure actions**, click on the checkbox next to **ContactMenuItem**.
3. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
4. Under **Configure actions**, click on the checkbox next to **AboutMenuItem**.
5. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
6. Click on **OK**.

Under the **OnWindowScroll** event, right-click on the **WindowScroll Y < 400** case and click **Toggle IF/ELSE IF**.

With the **WindowScroll Y < 400** case defined, we are now ready to enable the **WindowScroll Y > 400** case.

Enabling the WindowScroll Y > 400 & Y < 800 case

In the **Page Properties** pane under the design area, click on the **Page Interactions** tab. Double-click on the **OnWindowScroll** event. A **Case Editor** dialog box will open. In the **Case description** field, type `WindowScroll Y > 400 & Y < 800`. In the **Case Editor** dialog, do the following steps:

1. Add the first condition. Click on the **Add Condition** button. In the **Condition Builder** dialog box, go to the outlined condition box and perform the following steps:

    1. In the first dropdown, select **value**.
    2. In the second dropdown, select **[[Window.scrollY]]**.
    3. In the third dropdown, select **is greater than**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth dropdown, select **400**.

2. Add the second condition. Click on the green **+** sign. In the **Condition Builder** dialog box, go to the outlined condition box and perform these steps:

    1. In the first dropdown, select **value**.
    2. In the second dropdown, select **[[Window.scrollY]]**.
    3. In the third dropdown, select **is less than**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth dropdown, select **800**.
    6. Click on **OK**.

3. Create the first action. To set **Selected** to **True** for the **ContactMenuItem**, do as follows:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **ContactMenuItem**.
    3. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **true**.

4. Create the second action. To set **Selected** to **False** for the **WorkMenuItem** and **AboutMenuItem**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click the checkbox next to **WorkMenuItem**.
    3. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
    4. Under **Configure actions**, click on the checkbox next to **AboutMenuItem**.
    5. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
    6. Click on **OK**.

With the **WindowScroll Y > 400** case defined, we are now ready to create the **WindowScroll Y > 800** case.

Creating the WindowScroll Y > 800 case

In the **Page Properties** pane under the design area, click on the **Page Interactions** tab. Double-click the **OnWindowScroll** event. A **Case Editor** dialog box will open. In the **Case description** field, type `WindowScroll Y > 800`. In the **Case Editor** dialog, do the following:

1. Add the condition. Click on the **Add Condition** button. In the **Condition Builder** dialog box, go to the outlined condition box and perform the following steps:

    1. In the first dropdown, select **value**.
    2. In the second dropdown, select **[[Window.scrollY]]**.
    3. In the third dropdown, select **is greater than**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth dropdown, select **800**.

2. Create the first action. To set **Selected** to **True** for the **AboutMenuItem**, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click the checkbox next to **AboutMenuItem**.
    3. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **true**.

3. Create the second action. To set selected to **False** for the **WorkMenuItem** and **ContactMenuItem**, perform these steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Under the **Set Selected/Checked** drop-down menu, click on **Selected**.
    2. Under **Configure actions**, click on the checkbox next to **WorkMenuItem**.

3. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
4. Under **Configure actions**, click on the checkbox next to **ContactMenuItem**.
5. Under **Set selected state to**, verify that the first dropdown is set to **value** and second dropdown is set to **false**.
6. Click on **OK**.

We have now completed our portfolio site with a parallax scrolling effect.

Summary

In this chapter, we created a sample portfolio site with a parallax scrolling effect. Our site included left navigation. When the user scrolled past the given vertical scroll points, the left navigation updated to represent the section that was currently displayed. Clicking on a left navigation menu item, advanced the page to the corresponding section's page anchor. The **OnWindowScroll** page event controlled and enabled the parallax scrolling effect when the window was scrolled in the *y* direction.

In the next chapter, we will explore building a sample eBrochure page.

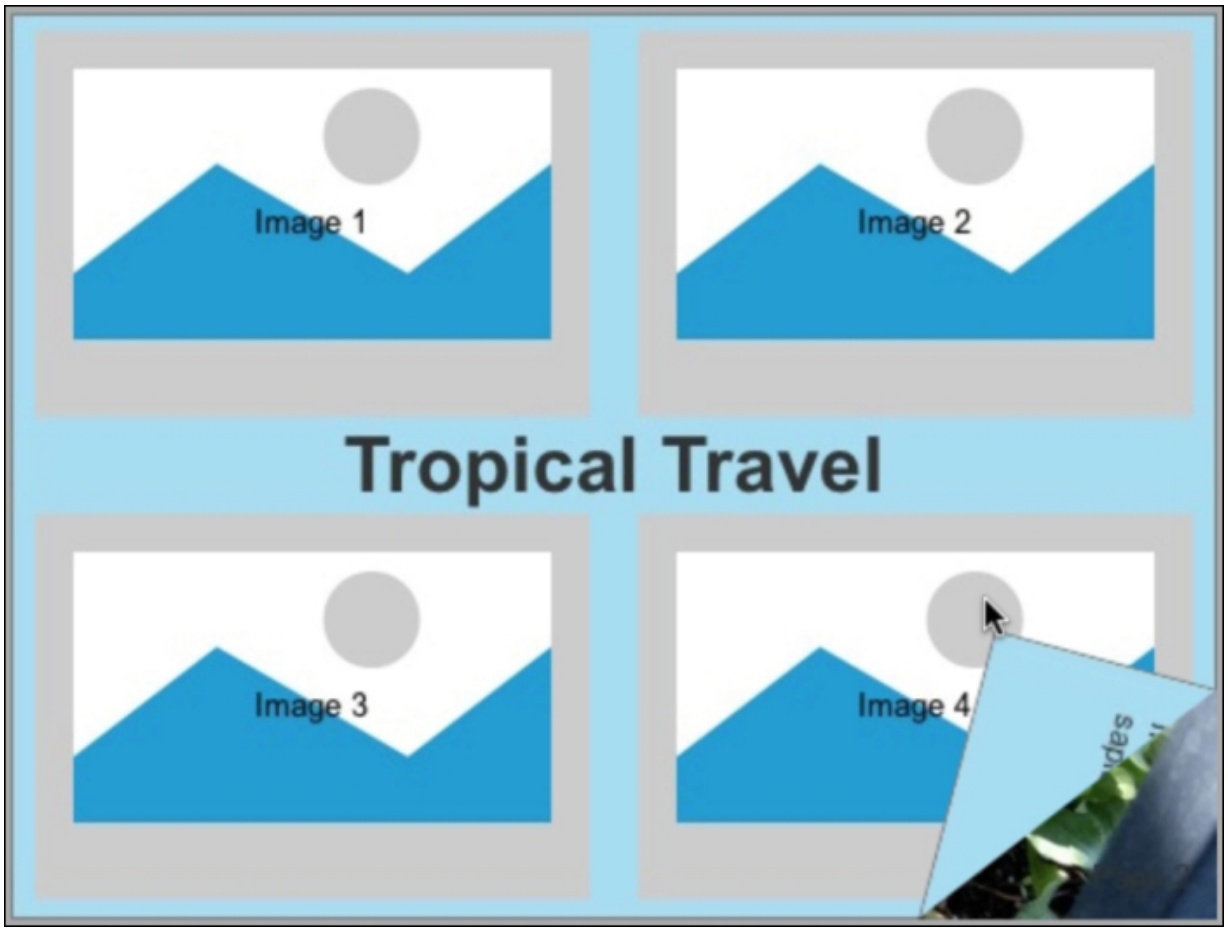
Chapter 6. Creating an eBrochure

With the hurried adoption of digital media, traditional media forms are rapidly evolving. Marketers, who once leveraged printed brochures exclusively, now replicate these materials online. Page flipping interactions provide a visceral connection between physical printed brochures and their digital representations.

We will explore how to create an eBrochure for a fictitious travel site using Axure, an external open source page flipper called jPageFlipper, and AxShare plugins.

> ## Note
>
> jPageFlipper was created by Ivan Suhinin, and is available for free under the MIT License. Learn more at https://jpageflipper.codeplex.com/.
>
> AxShare is Axure's cloud-based hosting service. You can host up to 1,000 prototypes for free. Learn more at http://share.axure.com/.

The following image shows what a user will experience when he/she starts to turn the cover of our eBrochure:



In this chapter, we will learn how to:

- Create our eBrochure page
    - Design content with Axure for our eBrochure
    - Convert pages into images
- Create a page flipper AxShare plugin

Creating our eBrochure page

For our open source page flipper to function properly, we will need to create a FlipperDP on the eBrochure page. After we have completed our prototype, we will create an AxShare plugin that will reference

the **FlipperDP** dynamic panel. The AxShare plugin will enable the jPageFlipper code to run in our prototype.

To design the eBrochure page, perform the following steps:

1. In the sitemap, double-click on the **ebrochure** page to open it in the design area.
2. From the **Widgets** pane, drag the **Dynamic Panel** widget  and place it at the coordinates (0,0). In the toolbar, change **w:** to `1200` and **h:** to `875`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `FlipperDP`.

We are now ready to design content for our eBrochure page.

Designing content with Axure for our eBrochure

We will start by creating varying types of page content using a combination of Axure and external images. Each page of our eBrochure will also be converted into an image file in the default jPageFlipper size of 500 by 375 pixels. Our eBrochure is comprised of six pages as follows:

Page title	**Description**
Cover	Created using standard Axure widgets
Page 1	Created using standard Axure widgets
Page 2	Created using standard Axure widgets with an external image imported.

Page title	Description
Page 3 – Left Side Image	Created using standard Axure widgets with a full panoramic image imported that spans two pages
Page 4 – Right Side Image	
Back	Created using standard Axure widgets

Designing our cover page

To design the cover page of our eBrochure, perform the following steps:

1. Open a new Axure document and organize the sitemap as follows:

    - **ebrochure**
    - **cover**
    - **page1**
    - **page2**
    - **page3**
    - **page4**
    - **back**
    - **snapshots**

2. In the sitemap, double-click on the **cover** page to open in the design area.

3. From the **Widgets** pane, drag the **Rectangle** widget  and place it at the coordinates (0,0).
4. Type `Tropical Travel`. In the toolbar, change **w:** to `500` and **h:** to `375`.
5. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Font** and perform the following steps:

    1. Click on the typeface dropdown and select **Bold**. Change the font size to `32`.
    2. Click on the down arrow next to the Text Color icon . In the drop-down menu, inside the **#** text field, enter `333333`.

6. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Borders, Lines, + Fills** and perform the following steps:

    1. Click on the down arrow next to the Line Color icon . In the drop-down menu, inside the **#** text field, enter `797979`.
    2. Click on the down arrow next to the Fill Color icon . In the drop-down menu, inside the **#** text field, enter `A6DDF2`.

7. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Alignment + Padding** and click on the Align Center and Align Middle icons.
8. Repeat step 3 using the values given in the following table to complete the design of the cover page (* If applicable):

Widget	**Coordinates**	**Text*** **(Shown on Widget)**	**Width** **(w:)**	**Height** **(h:)**
 Image	(10,7)	`Image 1`	`230`	`160`

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)
 Image	(260,7)	`Image 2`	`230`	`160`
 Image	(10,207)	`Image 3`	`230`	`160`
 Image	(260,207)	`Image 4`	`230`	`160`

With the cover page completed, we are now ready to design **page1**.

Designing page1

To design **page1** of our eBrochure, perform the following steps:

1. In the sitemap, double-click on the **page1** page to open in the design area.
2. From the **Widgets** pane, drag the **Rectangle** widget  and place it at the coordinates (0,0).
3. In the toolbar, change **w:** to `500` and **h:** to `375`.
4. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Font** and perform the following steps:

1. Click on the typeface dropdown and select **Bold**. Change the font size to `32`.
2. Click on the down arrow next to the Text Color icon . In the drop-down menu, inside the **#** text field, enter `333333`.

5. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Borders, Lines, + Fills** and perform the following steps:

    1. Click on the down arrow next to the Line Color icon . In the drop-down menu, inside the **#** text field, enter `797979`.
    2. Click on the down arrow next to the Fill Color icon . In the drop-down menu, in the **#** text field, enter `A6DDF2`.

6. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Alignment + Padding** and perform the following steps:

    1. Click on the Align Left and Align Top icons.
    2. Change padding **T** to `20`.

7. In the **Widgets**pane, drag the **Paragraph** widget  and place it at the coordinates (20,65). In the toolbar, change **w:** to `154` and **h:** to `255`. Adjust the **Paragraph** text to fit within **Page1Background**.
8. In the **Widgets** pane, drag the **Placeholder** widget  and place it at the coordinates (220,70). In the toolbar, change **w:** to `230` and **h:** to `245`.

With **page1** completed, we are now ready to design **page2**.

Designing page2

To design **page2** of our eBrochure, perform the following steps:

1. In the sitemap, double-click on the **page2** page to open in the design area.

2. From the **Widgets** pane, drag the **Image** widget  and place it at the coordinates (0,0).
3. In the toolbar, change **w:** to `500` and **h:** to `375`.

> ## Note
>
> If you wish to import an image with a width of 500 and height of 375, double-click on the **Image** widget at the coordinates (0,0) and open the appropriate image.

4. From the **Widgets** pane, drag the **Rectangle** widget  and place it at the coordinates (320,10).
5. In the toolbar, change **w:** to `170` and **h:** to `120`.
6. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Borders, Lines, + Fills** and perform the following steps:

    1. Click on the down arrow next to the Line Color icon . In the drop-down menu, in the **#** text field, enter `797979`.
    2. Click on the down arrow next to the Fill Color icon . In the drop-down menu, in the **#** text field, enter `FFFFFF`.
    3. In the **Opacity** field, enter `70`.

7. In the **Widgets** pane, drag the **Heading 1** widget  and place it at the coordinates (328,10).
8. In the **Widgets** pane, drag the **Paragraph** widget  and place it at the coordinates (328,50). In the toolbar, change **w:** to `154` and **h:** to `75`. Adjust the **Paragraph** text to fit within **HeroBackground**.

With **page2** completed, we are now ready to design **page3** using the left-hand side of our panoramic image.

Designing page3 using the left-hand side of our panoramic image

Select a panoramic image cropped to 1000 x 375 pixels and divide it into two images, each 500 pixels wide. To create **page3** using the left-hand side of our panoramic image, perform the following steps:

1. In the sitemap, double-click on the **page3** page to open in the design area.
2. From the **Widgets** pane, drag the **Image** widget  and place it at the coordinates (0,0). Right-click on the Image widget and click **Import Image…**, selecting the left-hand side of our panoramic image.
3. From the **Widgets** pane, drag the **Rectangle** widget  and place it at the coordinates (10,10).
4. In the toolbar, change **w:** to `390` and **h:** to `80`.
5. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Borders, Lines, + Fills** and perform the following steps:
    1. Click on the down arrow next to the Line Color icon . In the drop-down menu, in the **#** text field, enter `797979`.
    2. Click on the down arrow next to the Fill Color icon . In the drop-down menu, in the **#** text field, enter `FFFFFF`.
    3. In the **Opacity** field, enter `70`.
6. From the **Widgets** pane, drag the **Heading 1** widget  and place it at the coordinates (20,15). Type `J Your Private Getaway!`.
7. From the **Widgets** pane, drag the **Paragraph** widget  and place it at the coordinates (57,55). In the toolbar, change **w:** to `340` and **h:** to `30`. Adjust the **Paragraph** text to fit within **HeroBackground** (that is, two lines of text).

With **page3** completed, we are now ready to design **page4** using the right-hand side of our panoramic image.

Designing page4 using the right-hand side of our panoramic image

To create **page4** using the right-hand side of our panoramic image, perform the following steps:

1. In the sitemap, double-click the **page4** page to open in the design area.
2. From the **Widgets** pane, drag the **Image** widget  and place it at the coordinates (0,0). Right-click on the Image widget and click on **Import Image…**, selecting the right-hand side of our panoramic image.

With **page4** completed, we are now ready to design the **back** page.

Designing our back page

To design the **back** page of our eBrochure, perform the following steps:

1. In the sitemap, double-click on the **back** page to open in the design area.
2. From the **Widgets** pane, drag the **Rectangle** widget  and place it at the coordinates (0,0).
3. Type `Tropical Travel`. In the toolbar, change **w:** to `500` and **h:** to `375`.
4. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Font** and perform the following steps:
    1. Click on the typeface dropdown and select **Bold**. Change the font size to `32`.
    2. Click on the down arrow next to the Text Color icon . In the drop-down menu, in the **#** text field, enter `333333`.
5. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Borders, Lines, + Fills** and perform the

following steps:

1. Click on the down arrow next to the Line Color icon . In the drop-down menu, in the **#** text field, enter `797979`.
2. Click on the down arrow next to the Fill Color icon . In the drop-down menu, in the **#** text field, enter `A6DDF2`.

6. In the **Widget Properties and Style** pane with the **Style** tab selected, scroll to **Alignment + Padding** and click on the Align Center and Align Middle icons.
7. Repeat step 2 using the following table to complete the design of the **back** page (* If applicable):

Widget	**Coordinates**	**Text*** **(Shown on Widget)**	**Width (w:)**	**Height (h:)**
H2 Heading 2	(128,183)	100 Relaxation Way Hometown, NE 68001	236	56
A_ Label	(128,249)	(888) 555-1212 email@test.net	195	15

With all of the main pages of the eBrochure completed, we will create an AxShare plugin for jPageFlipper.

Converting pages into images

Our open source page flipper, jPageFlipper, leverages external image files. As a result, we now need to create images of each of the six pages we designed. Once we have created the images, we still need to provide our Axure prototype with access to the images.

To provide access to the images, we will upload the images to an unrestricted share on a cloud-based storage system called Dropbox. You can create a free account at https://www.dropbox.com.

Note

Once you upload images to Dropbox, you will see a share link. Clicking on the share link next to a file provides a link to the file, you will want to change the first part of the URL to `dl.dropbox.com` and remove the last part of the URL to provide a direct link to the file.

(For example, change https://www.dropbox.com/s/1tsov5ag1nscvo2/back.png?dl=0 to https://dl.dropboxusercontent.com/s/1tsov5ag1nscvo2/back.png.)

To create images of each page and import them into our prototype, with the Axure file still open in the main toolbar, navigate to **File** | **Export All Pages to Image…** and save it to a temporary folder. Log in to your Dropbox account and drag the following six images:

- `cover.png`
- `page1.png`
- `page2.png`
- `page3.png`
- `page4.png`
- `page5.png`
- `back.png`

You will also want to download the files for jPageFlipper 0.9 from https://jpageflipper.codeplex.com/releases/view/45532 and save them to your Dropbox. You will want to copy all of the direct links to each of the six images and the following jPageFlipper files:

- `jpageflipper/style.css`
- `jpageflipper/lib/jquery-1.4.2.min.js`
- `jpageflipper/lib/jquery-1.4.1-vsdoc.js`
- `jpageflipper/javascripts/jquery.pageFlipper.js`

With the above information, you are now ready to publish your prototype to AxShare and add the jPageFlipper plugin.

Creating a page flipper AxShare plugin

We will first publish our prototype to AxShare. Next, we will create our AxShare plugin for jPageFlipper. The plugin for AxShare feature allows us to place snippets of code inside a specified dynamic panel.

Publishing to AxShare

With our Axure file still open, in the main toolbar navigate to **Publish** | **Publish to Axure Share**.

Note

If you wish only to publish the **ebrochure** page, please perform the following steps in the main menu:

Go to **Publish** | **Generate HTML Files…**.

Click on the **Pages** menu item and uncheck **Generate All Pages**. Click on the checkbox next to **ebrochure** and click the **Close** button.

Now, when you publish to AxShare, only the **ebrochure** page will be published.

1. If you do not have an account, you can create one by clicking on the **Create Account** button and completing the form.
2. If you already have an account, click on the **Existing Account** button and enter your e-mail and password.
3. Once your prototype has been published to AxShare, make a note of the associated AxShare ID.

4. We will now log in to AxShare by opening https://share.axure.com in a web browser.
5. Click on the associated workspace and look for the corresponding AxShare ID. Click on the name of the project.
6. You will see the project details page open. Click on **Plugins** and then on **New Plugin**. In the **PluginName** field, enter `jPageFlipper`.
7. Under **Location**, click on the radio button next to **Inside Dynamic Panels with Name** and type `FlipperDP`.
8. In the content area, enter the following code replacing the highlighted areas with your appropriate Dropbox direct links:

```
<div>
        <link rel="stylesheet"
href="https://dl.dropboxusercontent.com/u/7235
888/jpageflipper/style.css" />
        <ul id="lstImages"
class="imagesSource">
          <li><img
src="https://dl.dropboxusercontent.com/s/syic7
2bwmu26u0z/cover.png" /></li>
          <li><img
src="https://dl.dropboxusercontent.com/s/icz2b
dpzlxlod49/page1.png" /></li>
          <li><img
src="https://dl.dropboxusercontent.com/s/qb6q1
4r90bmrl6w/page2.png" /></li>
          <li><img
src="https://dl.dropboxusercontent.com/s/4bqr6
r7hp9b6q4h/page3.png" /></li>
          <li><img
src="https://dl.dropboxusercontent.com/s/0wxcm
5gzteppq90/page4.png" /></li>
          <li><img
src="https://dl.dropboxusercontent.com/s/1tsov
5ag1nscvo2/back.png" /></li>
        </ul>

        <script type="text/javascript"
src="https://dl.dropboxusercontent.com/u/72358
88/jpageflipper/lib/jquery-1.4.2.min.js">
</script>
        <script type="text/javascript"
src="https://dl.dropboxusercontent.com/u/72358
88/jpageflipper/lib/jquery-1.4.1-vsdoc.js">
```

```
</script>
        <script type="text/javascript"
src="https://dl.dropboxusercontent.com/u/7235888/jpageflipper/javascripts/jquery.pageFlipper.js"></script>
        <script type="text/javascript">
          $(document).ready(function () {
            var isIPad =
navigator.userAgent.indexOf('iPad') >= 0;

            $('#lstImages').pageFlipper({
              fps: isIPad ? 10 : 20,
              easing: isIPad ? 0.3 : 0.2,
              backgroundColor: '#aaaaaa'
            });

            $('.canvasHolder').css('left',
(isIPad ? 0 : 130) + 'px');
            $('#mouse').css({
              width: (isIPad ? 40 : 20) +
'px',
              height: (isIPad ? 40 : 20) +
'px',
              '-moz-border-radius': (isIPad ?
20 : 10) + 'px',
              '-webkit-border-radius': (isIPad
? 20 : 10) + 'px'
            });
          });
        </script>
</div>
```

9. Click **Save and Close**. On the project details page, click on the link to the project to view the prototype.

Summary

We created a travel eBrochure that leveraged an external JavaScript page flipping effect. To accomplish this, we used Axure to design our pages and then generated image files of all the pages. We then uploaded the image files, as well as the necessary files for the JavaScript page flipper, to our Dropbox share. We created direct links to all of the content in Dropbox and then published the prototype to AxShare. We logged in to AxShare and then created a plugin that referenced our external custom CSS, JavaScript, and image files.

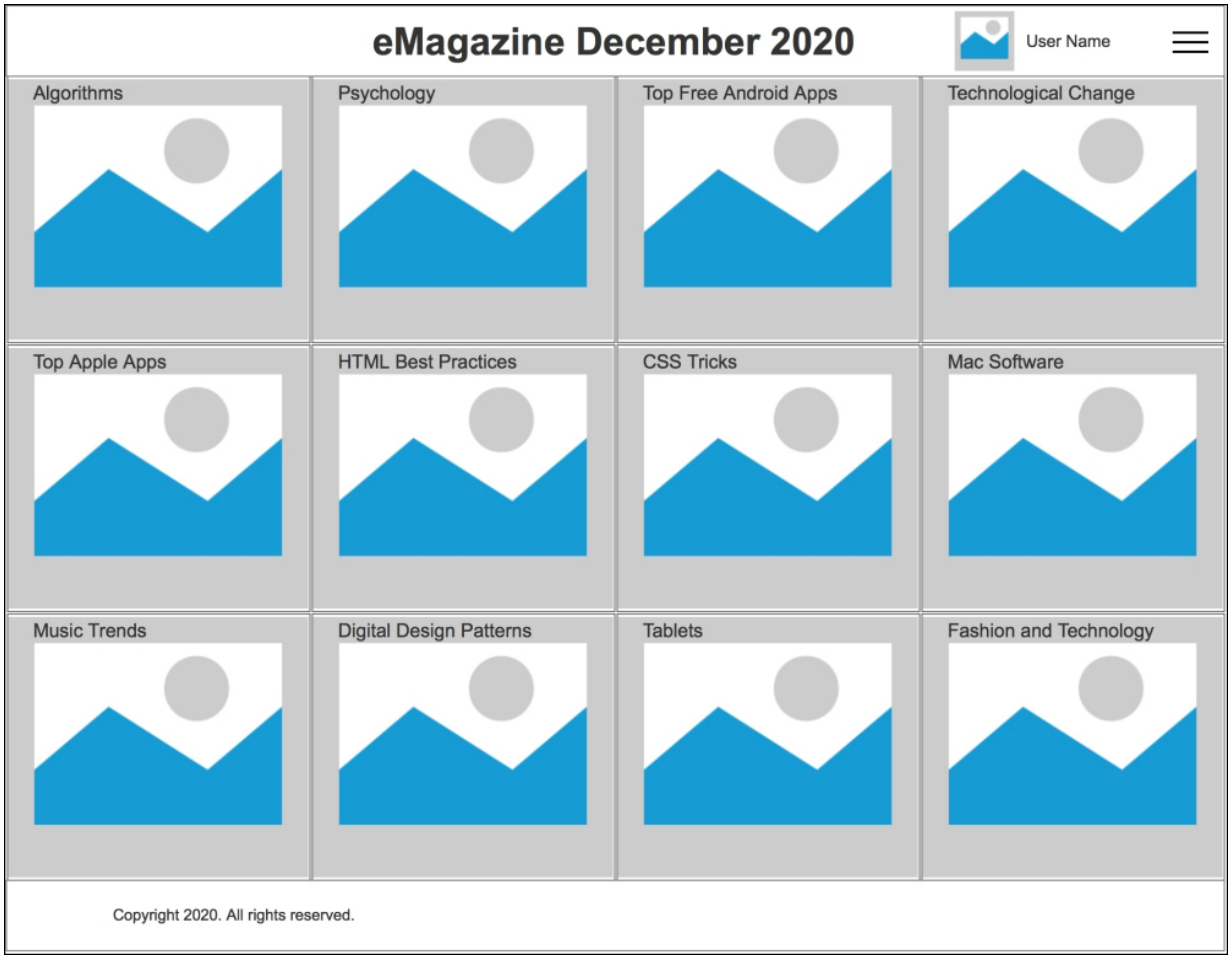
In the next chapter, we will explore building a sample eMagazine optimized for tablet and mobile interactions.

Chapter 7. Creating an eMagazine

The Adaptive View feature released in Axure RP 7 allows us to create specific breakpoints optimized for various mobile devices. To explore Adaptive views further, we will create an eMagazine that will be tailored for the iPad and iPhone. To accomplish this, we will leverage two Adaptive views:

- Portrait tablet (768 x any and below)
- Portrait phone (320 x any and below)

Using this configuration, we will design our Base view, targeting a Landscape Tablet view of 1,024 pixels (that is, optimized to be backwards compatible with the original iPad). The home page of our eMagazine will look as follows:



In this chapter, we will cover the following topics:

- Designing an eMagazine

- Creating eMagazine interactions

Designing an eMagazine

For our eMagazine, we will have three pages as follows:

- eMagazine Home
- Topic Overview
- Article Detail

We will start by renaming the pages in the sitemap. We will then configure the Adaptive views. Next, we will create a global header and footer that will be used by all pages. For the home page, will create an optimized repeater widget, designed to nicely adapt to each of our targeted viewports. The **Topic Overview** page will use three zones to showcase the featured articles. Each zone will comprise of Image, Heading 2, Label, Shape, and Paragraph widgets. Finally, the **Article Detail** page will contain an Image, Heading 2, Label, Shape, and five Paragraph widgets.

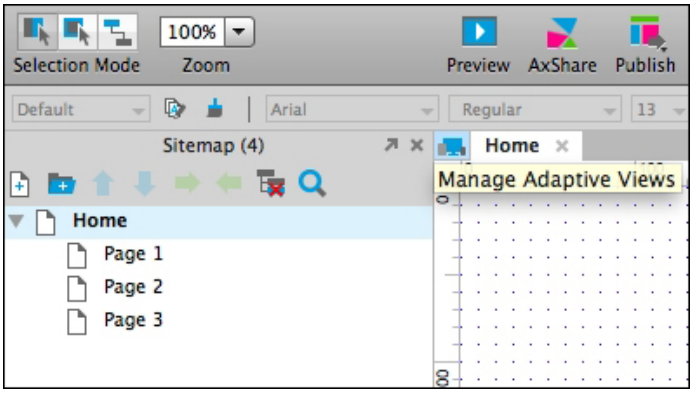
Updating the sitemap and configuring Adaptive Views

To update the sitemap and configure the Adaptive Views for the eMagazine, perform the following steps:

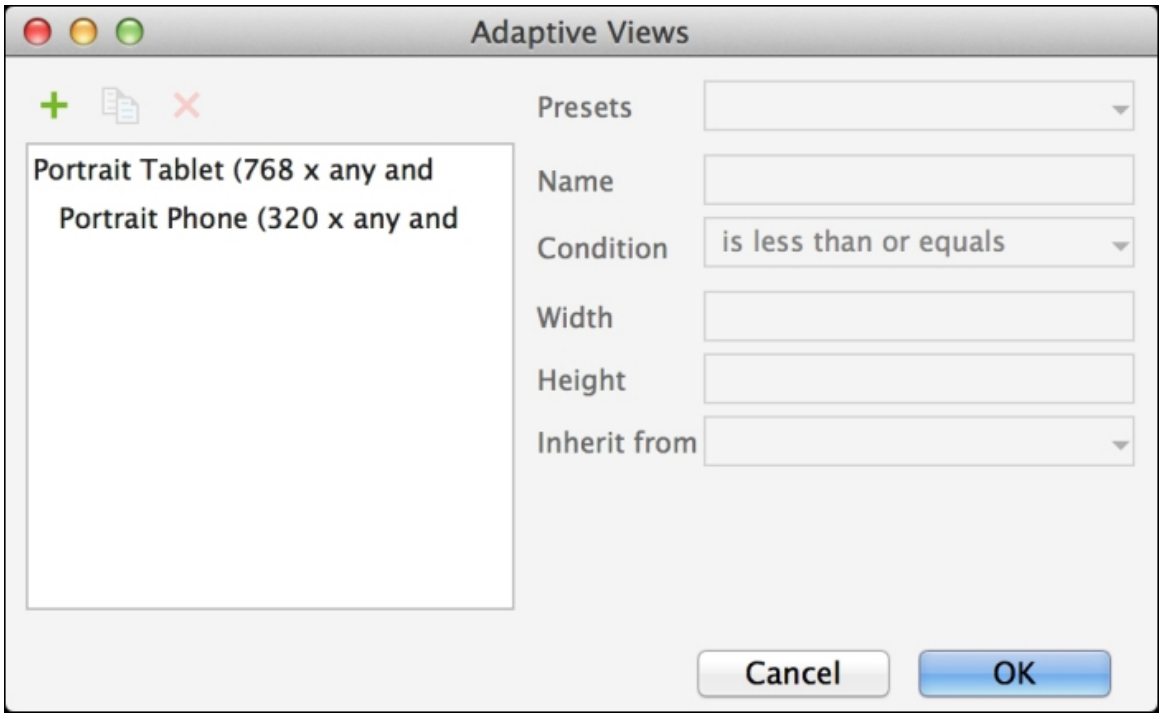
1. Open a new Axure document and organize the sitemap as follows:

    - eMagazine Home
    - Topic Overview
    - Article Detail

2. To configure Adaptive Views, click on the **Manage Adaptive Views** icon  that is located on the upper-left corner of the design area, as shown in the following screenshot:



3. In the **Adaptive Views** dialog box, click on the green plus sign, and from the **Presets** dropdown menu, select **Portrait Tablet**.
4. Click again on the green plus sign and from the **Presets** dropdown menu, select **Portrait Phone**. Click on **OK**. Once completed, your **Adaptive Views** dialog box should look like the following screenshot:



With our sitemap updated and the adaptive views configured, we will create global variables as well as the design for our **Header and Footer** masters.

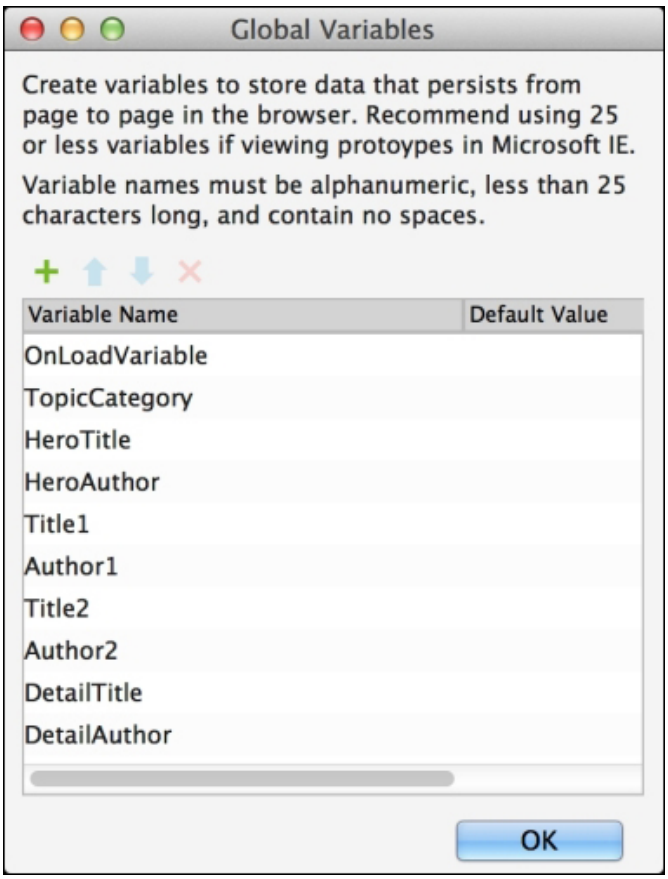
Creating global variables

We will now create global variables, which will allow us share data between pages in our design. To create global variables, in the main menu, click on **Project** and then **Global Variables…**. In the **Global Variables** dialog, perform the following steps:

1. Click on the green **+** sign and type `TopicCategory`. We will leave the **Default Value** field empty.
2. Repeat Step 1 eight more times to create additional variables using the following table for **Variable Name** field:

Variable Name
HeroTitle
HeroAuthor
Title1
Author1
Title2
Author2
DetailTitle
DetailAuthor

Once completed, your **Global Variables** dialog box will look like this:



3. Click on **OK**.

With our global variables created, we are now ready to complete the design and interactions for the Header and Footer masters. We will start with adding masters to the **Masters** pane.

Adding masters to the Masters pane

To create masters, perform the following steps:

1. In the **Masters** pane, click on the **Add Master** icon, type `Header`, and press *Enter*.
2. Next, right-click on the icon next to the **Header** master, mouse over **Drop Behavior**, and click **Lock to Master Location**.
3. Again, in the **Masters** pane, click on the **Add Master** icon, type `Footer` and press *Enter*.

With our Masters added to the Masters pane, we are now ready to complete the design and interactions for each individual Master. We will start with the Header master.

Designing the Header master

The **Header** master will be used on all pages of our eMagazine as identified in our sitemap. Once completed, the **Header** master will look as follows:



To place widgets on the **Header** master, select the **Base** Adaptive View and perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Header** master to open in the design area.
2. From the **Widgets** pane, drag the **Rectangle** widget and place it at coordinates (0,0). In the toolbar, change **w:**`1024` and **h:**`60`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `HeaderBackground`.
3. Repeat Step 2 to complete the Base view of the **Header** master using the following table (* if applicable):

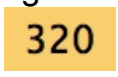
Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
H1 Placeholder	(308,12)	`eMagazine December 2020`	`394`	`37`	`eMagazineIssue`
Image	(797,5)		`50`	`50`	`UserImage`
Label	(857,22)	`Username`	`73`	`16`	`UserNameLabel`

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
 Horizontal Line	(980,18)		30		MenuPartA
 Horizontal Line	(980,26)		30		MenuPartB
 Horizontal Line	(980,34)		30		MenuPartC

To place widgets on the **Header** master in the Portrait Tablet view, click on the 768 button  that is located in the upper-left corner of the design area. With the **Header** master open in the design area, perform the following steps:

4. Reposition and resize widgets to complete the 768 view using the following table (* if applicable):

Named Widget (In the Widget Interactions and Notes pane)	Base View Coordinates	New Coordinates*	New Width* (w:)	New Height* (h:)	New Font*
HeaderBackground	(0,0)		768		
eMagazineIssue	(308,12)	(180,14)	345	32	28
UserImage	(797,5)	(557,5)			
UserNameLabel	(857,22)	(617,22)			
MenuPartA	(980,18)	(720,18)			
MenuPartB	(980,26)	(720,26)			
MenuPartC	(980,34)	(720,34)			

To place widgets on the **Header** master on the Portrait tablet view, click on the 320 button 

that is located in the upper left corner of the design area. With the **Header** master opened in the design area, perform the following steps:

1. Reposition and resize widgets to complete the 320 view using the following table (* if applicable):

Named Widget (In the Widget Interactions and Notes pane)	768 View Coordinates	New Coordinates*	New Width* (w:)	New Height* (h:)	New Font*
HeaderBackground	(0,0)		320	60	
eMagazineIssue	(180,14)	(70,9)	129	42	18
UserImage	(557,5)	(222,6)	33	33	
UserNameLabel	(617,22)	(215,43)	52	11	10
MenuPartA	(720,18)	(280,17)			
MenuPartB	(720,26)	(280,25)			
MenuPartC	(720,34)	(280,33)			

We will now design our **Footer** master.

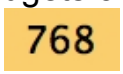
Designing the Footer master

The **Footer** master will be used on all pages of our eMagazine, as identified in our **sitemap**. Once completed, the **Footer** master will look as follows:



To place widgets on the **Footer** master, select the **Base** Adaptive View and perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Header** master to open in the design area.
2. From the **Widgets** pane, drag the **Rectangle** widget and place it at coordinates (0,0). In the toolbar change **w:** `1024` and **h:** `60`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `FooterBackground`.
3. From the **Widgets** pane, drag the **Label** widget and place at coordinates (90,22). In this widget, type `Copyright 2020. All rights reserved`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `FooterCopyright`.

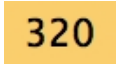
To place widgets on the **Footer** master on the Portrait Tablet view, click on the 768 button 

that is located in the upper-left corner of the design area, and perform the following step:

1. Reposition and resize widgets to complete the 768 view using the following table (* if applicable):

Named Widget **(In the Widget Interactions and Notes pane)**	**Base View Coordinates**	**New Coordinates***	**New Width*** **(w:)**	**New Height*** **(h:)**
FooterBackground	(0,0)		`768`	

> ## Note
>
> The **FooterCopyright** widget does not need to be repositioned or resized for the 768 view.

To place widgets on the Footer master in the Portrait Tablet view, click on the 320 button 

that is located in the upper-left corner of the design area and perform the following step:

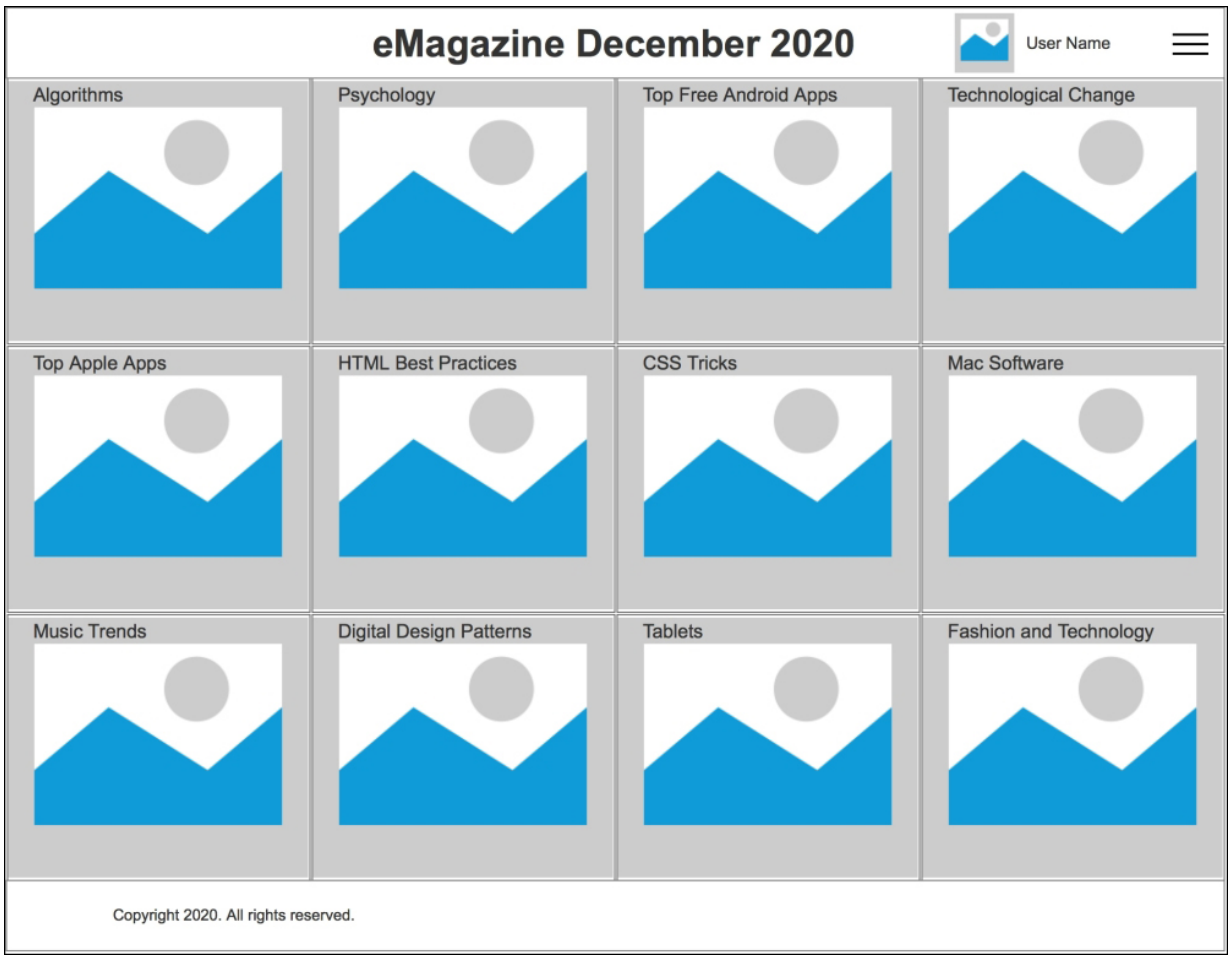
1. Reposition and resize widgets to complete the 320 view using the following table (* if applicable):

Named Widget **(In the Widget Interactions and Notes pane)**	**768 View Coordinates**	**New Coordinates***	**New Width*** **(w:)**	**New Height*** **(h:)**
FooterBackground	(0,0)		320	
FooterCopyright	(90,22)	(56,22)		

With the **Header** and **Footer** masters completed, we are now ready to design the home page.

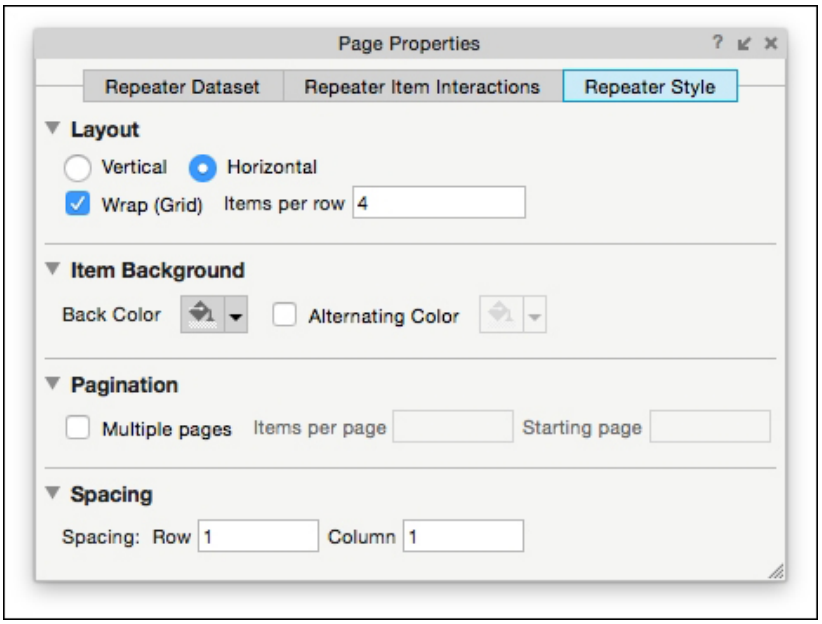
Designing the eMagazine home page

The content for the eMagazine home page is provided by an adaptive Repeater widget optimized for each adaptive view. Once completed, the home page will look as follows:



To design the home page for our eMagazine, select the **Base** Adaptive View and perform the following steps:

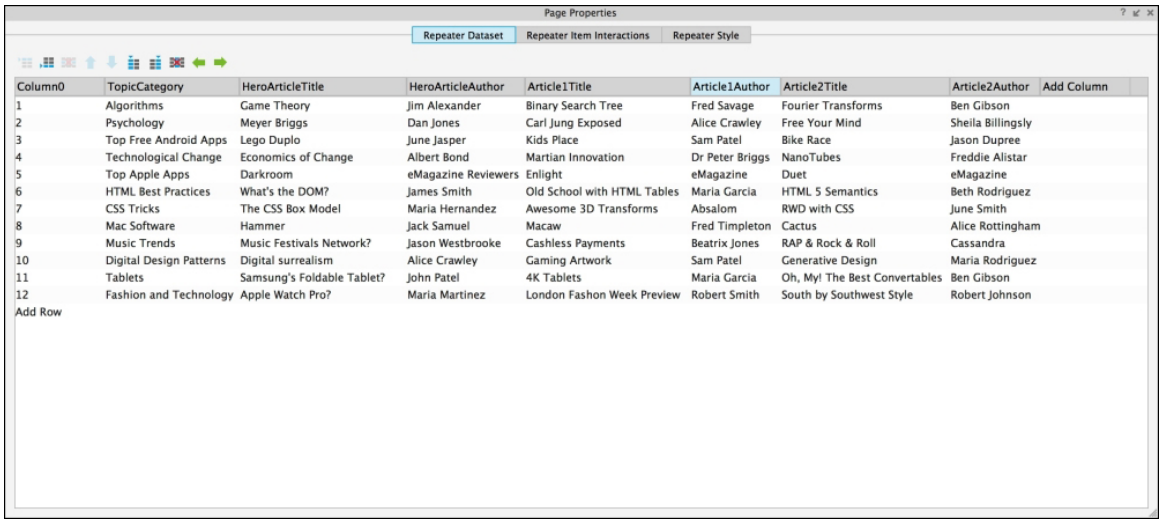
1. In the sitemap, double-click on eMagazine home page to open it in the design area.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. From the **Masters** pane, drag the **Footer** master and drop it at coordinates (0,732).
4. From the **Widgets** pane, drag the **Repeater** widget into the design area and place at coordinates (1,59). In the **Widget Interactions and Notes** pane, click on the **RepeaterName** field and type `eMagazineHomeRepeater`.
5. Double-click on the **eMagazineHomeRepeater Repeater** widget to open it in the design area.
6. Under the design area in the **Repeater** pane, click on the **Repeater Style** tab. Once updated, your **Repeater Style** tab will look as follows:



To change the default **Repeater Style**, perform the following steps:

1. Under the **Layout** dropdown, click on the radio button next to **Horizontal**.
2. Select the checkbox next to **Wrap (Grid)**, and in the **Items per row** field, enter `4`.
3. Under the **Spacing** dropdown, enter `1` in both the **Row** and **Column** fields.
4. Under the design area in the **Repeater** pane, click on the **Repeater Dataset** tab. Use the following table to update data values:

> ## Note
>
> To rename columns or rows, double-click on the column or row and type the new column or row name. To add new columns or rows, click on **Add column** or **add row**.



We will now resize the Rectangle widget on the Repeater Item and place additional widgets. Perform the following steps:

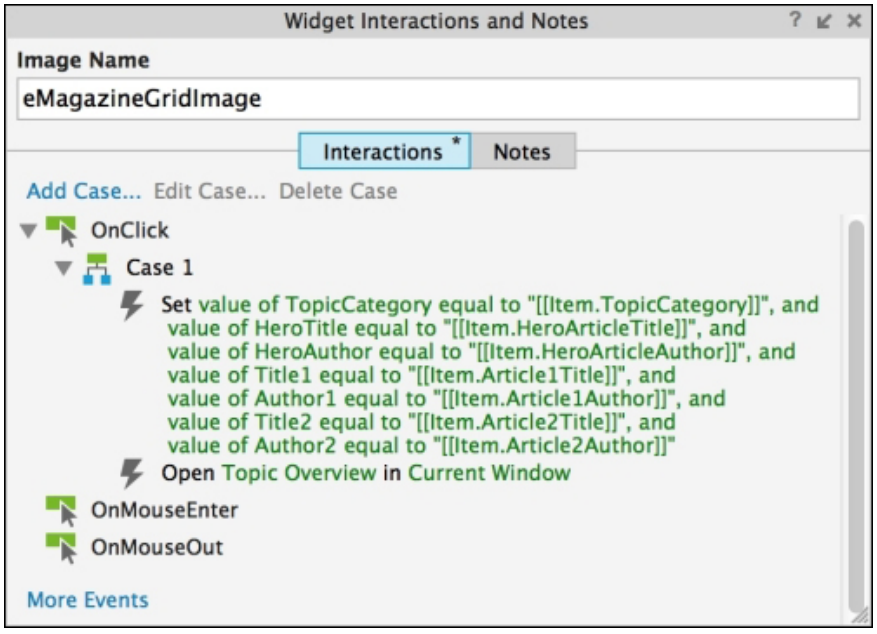
1. With the Repeater still opened in the design area, click on the **Rectangle** widget at coordinates (0,0). In the toolbar, change **w:** `255` and **h:** `224`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Background`.

> ## Note
>
> Since we chose a width of 255 for the Background widget, we will only have to adjust the **Spacing: Row** field for each Adaptive View and will not have to change the Repeater Item for our 768 and 320 views.

2. From the **Widgets** pane, drag the **Image** widget and place it at coordinates (1,3). In the toolbar, change **w:**`252` and **h:**`218`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `eMagazineGridImage`.
3. From the **Widgets** pane, drag the **Label** widget and place it at coordinates (22,5). In this widget, type `Topic Category`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `TopicCategoryLabel`.

Since we have already defined all of our global variables, we can now complete the interactions for our Repeater Item. All interactions on the Repeater Item are defined in the **OnClick** event for **eMagazineGridImage**. Once defined, the

Widget Interactions and Notes pane for the eMagazineGridImage widget will look as follows:



We will perform the following steps.

With the Repeater Item still opened in the design area and the **Base** Adaptive View selected, in the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click on the **OnClick** event. A **Case Editor** dialog box will open.

Create the first action. To Set Variable Value for key global variables, follow the subsequent steps:

1. Under **Click to add actions**, scroll to the **Widgets** dropdown menu, and click on **Set Variable Value**.
2. Under **Configure actions**, click on the checkbox next to **TopicCategory**.
3. Under **Set variable to**, click on the first dropdown and then click on **Value**. In the text field, enter `[[Item.TopicCategory]]`.
4. Repeat steps 1-3 using the following table for **Variables to set** and **Value**:

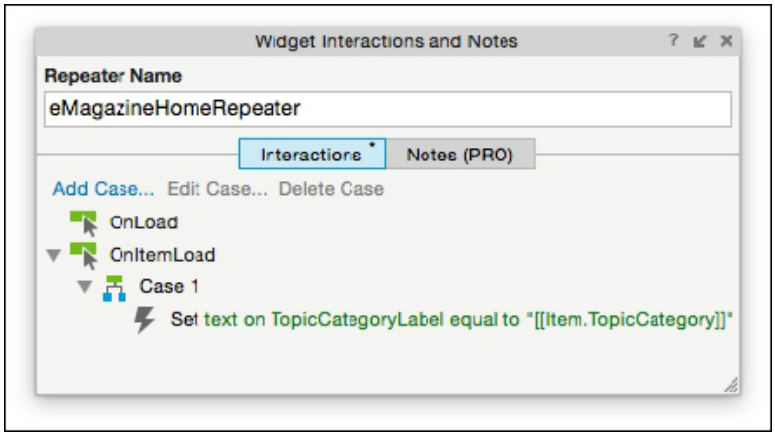
Variables to set (In the Widget Interactions and Notes pane)	Value
HeroTitle	`[[Item.HeroArticleTitle]]`
HeroAuthor	`[[Item.HeroArticleAuthor]]`
Title1	`[[Item.Article1Title]]`
Author1	`[[Item.Article1Author]]`
Title2	`[[Item.Article2Title]]`
Author2	`[[Item.Article2Author]]`

Create the second action. To open topic overview in the current window, do the following:

1. Under **Click to add actions**, in the **Links** dropdown menu under the **Open Link** dropdown menu, click on **Current Window**.
2. Under **Configure actions**, click on **Topic Overview**, and then click on **OK**.

To complete interactions for the Repeater, we will also need to update the **OnItemLoad** event.

Once defined, the **OnItemLoad** event of `eMagazineHomeRepeater` will look as follows:



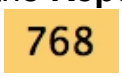
Perform the following steps.

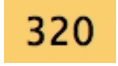
Under the design area in the **Repeater** pane, click on the **Repeater Item Interactions** tab. Double-click on the **OnItemLoad** event. A **Case Editor** dialog box will open. In the **Case Editor** dialog, do the following:

Create the first action. To set text on the **TopicCategoryLabel**, complete these steps:

1. Under **Click to add actions**, scroll to the **Widgets** dropdown menu, and click on **Set Text**.
2. Under **Configure actions**, click on the checkbox next to **TopicCategoryLabel**.
3. Under **Set text to**, click on the first dropdown and then on **Value**. In the text field, enter `[[Item.TopicCategory]]`.

With the Base view designed and the Dataset and interactions for the Repeater defined, we are now ready to update the **Repeater Spacing: Row** field for each Adaptive View for the 768 and 320 views as follows:

- To update the **Repeater Spacing: Row** field for the 768 view, click on the 768 button 

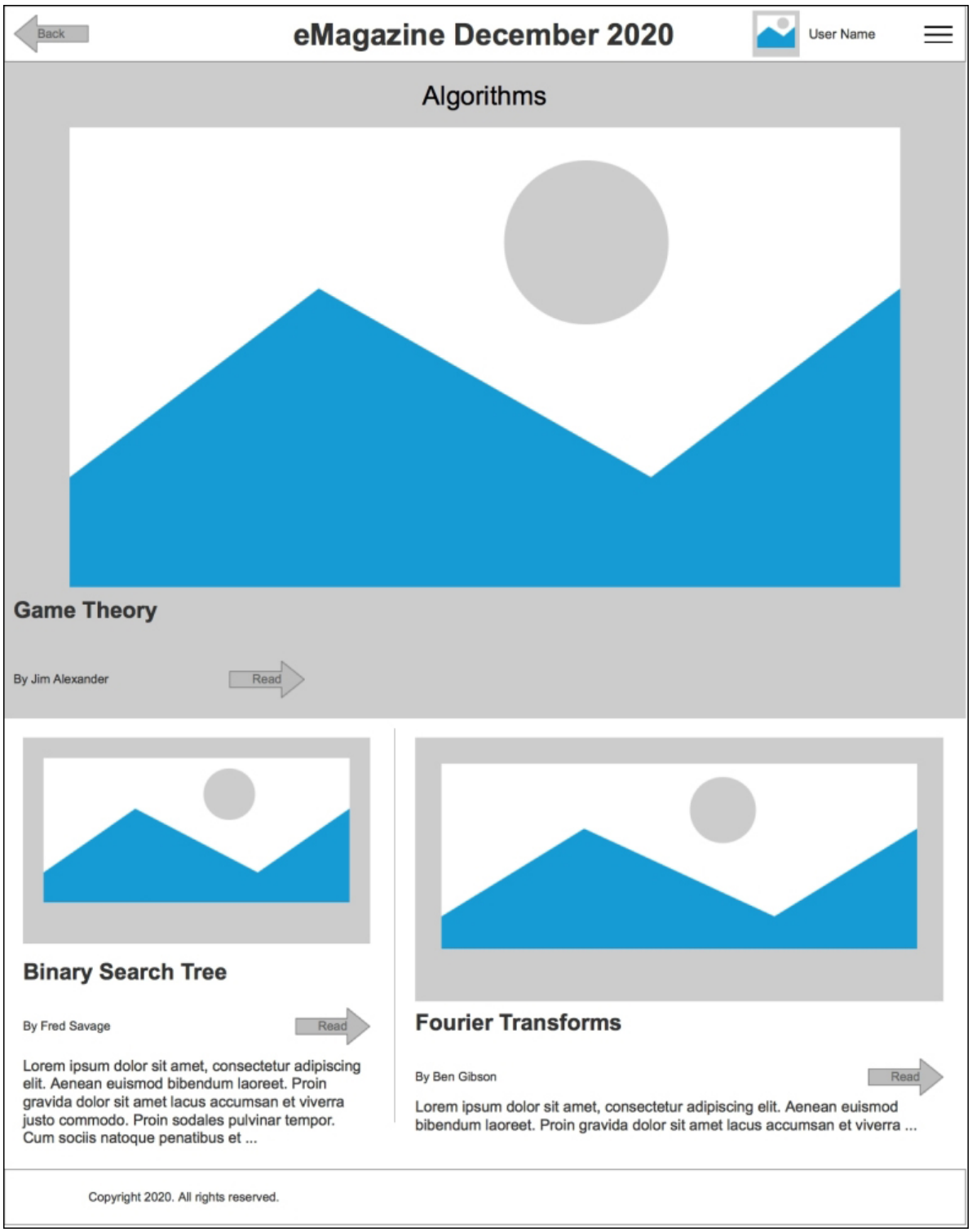
that is located in the upper-left corner of the design area. Under the design area in the **Repeater** pane, click on the **Repeater Style** tab. Under the **Layout** dropdown, click on the radio button next to **Horizontal**. In the **Items per row** field, enter `3`.
- To update the **Repeater Spacing: Row** field for the 320 view, click on the 320 button 

that is located in the upper-left corner of the design area. Under the design area in the **Repeater** pane, click the **Repeater Style** tab. Under the **Layout** dropdown, click on the radio button next to **Horizontal**. In the **Items per row** field, enter `1`.

With the design and interactions completed for the eMagazine home page, we are now ready to design the **Topic Overview** page.

Designing the Topic Overview page

Once completed, the **Topic Overview** page will look as follows:



The **Topic Overview** page will leverage the global header and footer and contain the following widgets:

Widget names	**Widget types**
BackArrow	Shape
HeroImage	Image
ArticleDetailTitle	Heading 2
ArticleDetailName1	Label
TopicHeroReadArrow	Shape
Article1Image	Image
Article1Title	Heading 2
Author1Name	Label
Article1ReadArrow	Shape
Article1Abstract	Paragraph
VerticalSeparator	Vertical Line
Article2Image	Image
Article2Title	Heading 2

Widget names	Widget types
Author2Name	Label
Article2ReadArrow	Shape
Article2Abstract	Paragraph

To design the **Topic Overview** page for our eMagazine, perform the following steps:

1. In the sitemap, double-click on the **Topic Overview** page to open in the design area. Select the **Base** Adaptive View.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. From the **Masters** pane, drag the **Footer** master and drop it at coordinates (0,1250).
4. From the **Widgets** pane, drag the **Rectangle** widget and drop it at coordinates (10,10). In the toolbar, change **w:** `80` and **h:** `40`. Type `Back`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `BackArrow`. Right-click on the **Rectangle** widget, mouse over **Select Shape**, and select **Arrow Left**.

   > ## Note
   >
   > If you choose, you can now adjust the Fill Color, Line Color, and Text Color of the Rectangle widget to blend with the other elements in the Header.

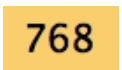
5. Repeat step 4 using the following table (* if applicable):

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
 Image	(0,60)	Topic (Note: Font 28; Padding T: 20)	1024	700	TopicHeroImage
H2 Heading 2	(10,630)	Article Title	310	28	HeroArticleTitle
A_ Label	(10,710)	By Author Name	97	15	HeroAuthorName
 Rectangle Widget (right arrow)	(240,698)	Read	80	40	TopicHeroReadArrow
 Image	(20,780)		370	220	Article1Image
H2 Heading 2	(20,1016)	Article Title	280	28	Article1Title

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
 Label	(20,1080)	By Author Name	97	15	Author1Name
 Rectangle Widget (right arrow)	(310,1068)	Read	80	40	Article1ReadArrow
 Paragraph	(20,1120)	(Note: Adjust Paragraph content to fit.)	370	90	Article1Abstract
 Vertical Line	(410,770)			450	VerticalSeparator
 Image	(438,780)		562	281	Article2Image

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
H2 Heading 2	(438,1080)	`Article Title`	472	28	Article2Title
Label	(438,1144)	`By Author Name`	97	15	Author2Name
Rectangle Widget (right arrow)	(920,1132)	`Read`	80	40	Article2ReadArrow
Paragraph	(438,1184)	(Note: Adjust Paragraph content to fit.)	562	36	Article2Abstract

With the Base view designed, we are now ready to update the design for each adaptive view for the 768 and 320 views.

To update the 768 view, click on the 768 button  that is located in the upper-left corner of the design area. With the **Topic Overview** page still open in the design area, perform the following steps:

1. In the design area, click on the **Footer** master. In the toolbar, change **x:** to `0` and **y:** to `1080`.

2. Reposition and resize widgets to complete the 768 view using the following table (* if applicable):

Named Widget **(In the Widget Interactions and Notes pane)**	**Base View Coordinates**	**New Coordinates***	**New Width*** **(w:)**	**New Height*** **(h:)**	**New Font***
TopicHeroImage	(0,60)		768	523	
HeroArticleTitle	(10,630)	(10,470)			
HeroAuthorName	(10,710)	(10,543)			
TopicHeroReadArrow	(240,698)	(240,530)			
Article1Image	(20,780)	(20,618)	282	168	
Article1Title	(20,1016)	(20,801)			
Author1Name	(20,1080)	(20,875)			
Article1ReadArrow	(310,1068)	(220,862)			
Article1Abstract	(20,1120)	(20,914)	290	126	
VerticalSeparator	(410,770)	(316,610)		410	
Article2Image	(438,780)	(339,618)	429	234	

Named Widget (In the Widget Interactions and Notes pane)	Base View Coordinates	New Coordinates*	New Width* (w:)	New Height* (h:)	New Font*
Article2Title	(438,1080)	(340,870)	280		
Author2Name	(438,1144)	(340,946)			
Article2ReadArrow	(920,1132)	(690,934)			
Article2Abstract	(438,1184)	(340,986)	428	54	

To update the 320 view, click on the 320 button 

that is located in the upper-left corner of the design area. With the Topic Overview page still open in the design area, perform the following steps:

1. In the design area, click on the **Footer** master. In the toolbar, change **x:** to 0 and **y:** to 1180.
2. Reposition and resize widgets to complete the 320 view using the following table (* if applicable):

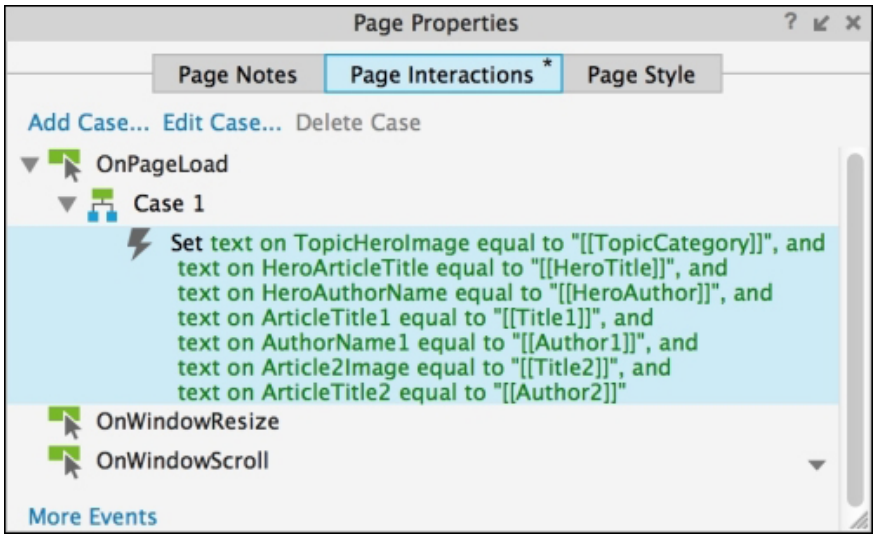
Named Widget (In the Widget Interactions and Notes pane)	768 View Coordinates	New Coordinates*	New Width* (w:)	New Height* (h:)	New Font*
BackArrow	(10,10)	(5,16)	50	30	

Named Widget (In the Widget Interactions and Notes pane)	768 View Coordinates	New Coordinates*	New Width* (w:)	New Height* (h:)	New Font*
TopicHeroImage	(0,60)		320	220	
HeroArticleTitle	(10,470)	(10,190)			
HeroAuthorName	(10,543)	(10,251)			
TopicHeroReadArrow	(240,530)	(240,244)	50	30	
Article1Image	(20,618)	(20,330)			
Article1Title	(20,801)	(20,513)			
Author1Name	(20,875)	(20,579)			
Article1ReadArrow	(220,862)	(252,572)	50	30	
Article1Abstract	(20,914)	(20,608)			
VerticalSeparator	(316,610)	Right-click and Unplace from View			
Article2Image	(339,618)	(10,808)	300	150	
Article2Title	(340,870)	(11,970)			

Named Widget (In the Widget Interactions and Notes pane)	768 View Coordinates	New Coordinates*	New Width* (w:)	New Height* (h:)	New Font*
Author2Name	(340,946)	(11,1048)			
Article2ReadArrow	(690,934)	(260,1040)	50	30	
Article2Abstract	(340,986)	(11,1078)	299	72	

Defining the OnPageLoad interaction

OnPageLoad of the **Topic Overview** page, we want to set text on our Image, Title, and Author Name widgets. Once completed, the **Page Properties** pane's **Page Interactions** tab will look as follows:



Under the design area in the Page Properties pane, click on the **Page Interactions** tab. Double-click on the **OnPageLoad** event. A **Case Editor** dialog box will open. In the **Case Editor** dialog, create the action. To Set Text on the widgets, perform the following steps:

1. Under **Click to add actions**, scroll to the **Widgets** dropdown menu, and click on **Set Text**.
2. Under **Configure actions**, click on the checkbox next to **TopicHeroImage**.
3. Under **Set text to**, click the first dropdown and then click **Value**. In the text field, enter `[[TopicCategory]]`.
4. Repeat steps 2 and 3 using the following table to select which **widgets** to Set text and **Value** for:

Variables to set	**Value**
HeroArticleTitle	`[[HeroTitle]]`
HeroAuthorName	`[[HeroAuthor]]`
Article1Title	`[[Title1]]`
Author1Name	`[[Author1]]`
Article2Title	`[[Title2]]`
Author2Name	`[[Author2]]`

5. Click on **OK**.

Defining OnClick interactions

Next, we will define the individual **OnClick** interactions for our **Topic Overview** page. To do so, perform the following steps:

1. With the **Topic Overview** page open in the design area, select the **Base** Adaptive View. Click on **BackArrow** at coordinates (10,10). To open e-commerce home in the current window, under **Click to add actions**, in the **Links** dropdown menu under the **Open Link** dropdown menu, click on

Current Window. Under **Configure actions**, click on **eCommerce Home**. Click **OK**.

2. Click on TopicHeroImage at coordinates (0,60). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click on **OnClick**. A **Case Editor** dialog box will open.
3. Create the first action. To set value of the **DetailTitle** and **DetailAuthor** variables, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown menu and click on **Set value**. Under **Configure actions**, click on the checkbox next to **DetailTitle**. Under **Set variable to**, click on the first dropdown and then click on **Value**. In the text field, enter `[[HeroTitle]]`.
    2. Under **Configure actions**, click on the checkbox next to **DetailAuthor**. Under **Set variable to**, click the first dropdown and then on **Value**. In the text field, enter `[[HeroAuthor]]`.

4. Create the second action. To open the **Article Detail** page in the Current Window, under **Click to add actions**, in the **Links** dropdown menu under the **Open Link** dropdown menu, click on **Current Window**. Under **Configure actions**, click on **Article Detail**. Click on **OK**.
5. With **TopicHeroImage** at coordinates (0,60) still selected, right-click on **Case 1** under the **OnClick** event and select **Copy**.
6. Click on **HeroArticleTitle** at coordinates (10,630). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, then right-click on **OnClick**, and select **Paste**.
7. Click on **TopicHeroReadArrow** at coordinates (240,698). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, then right-click on **OnClick**, and select **Paste**.
8. Click on **Article1Image** at coordinates (20,780). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click on **OnClick**. A **Case Editor** dialog box will open. In the **Case Editor** dialog, create the actions:
9. Create the first action. To set the value of the **DetailTitle** and **DetailAuthor** variables, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown menu and click on **Set value**. Under **Configure actions**, click on the checkbox next to **DetailTitle**. Under **Set variable to**, click on the first dropdown and then on **Value**. In the text field, enter `[[Title1]]`.
    2. Under **Configure actions**, click on the checkbox next to **DetailAuthor**. Under **Set variable to**, click on the first dropdown and then on **Value**. In the text field, enter `[[Author1]]`.

10. Create the second action. To open the **Article Detail** page in the current window, under **Click to add actions**, in the **Links** dropdown menu under the **Open Link** dropdown menu, click on **Current Window**. Under **Configure actions**, click on **Article Detail**. Click **OK**.
11. With **Article1Image** at coordinates (20,780) still selected, right-click on **Case 1** under the **OnClick** event and select **Copy**.
12. Click on **Article1Title** at coordinates (20,1016). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and right-click on **OnClick** and select **Paste**.
13. Click on **Article1ReadArrow** at coordinates (310,1068). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, right-click on **OnClick**, and select **Paste**.
14. Click on **Article2Image** at coordinates (438,780). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click **OnClick**. A **Case Editor** dialog box will open. In the **Case Editor** dialog, Create the actions:
15. Create the first action. To set the value of the DetailTitle and DetailAuthor variables, perform the following steps:

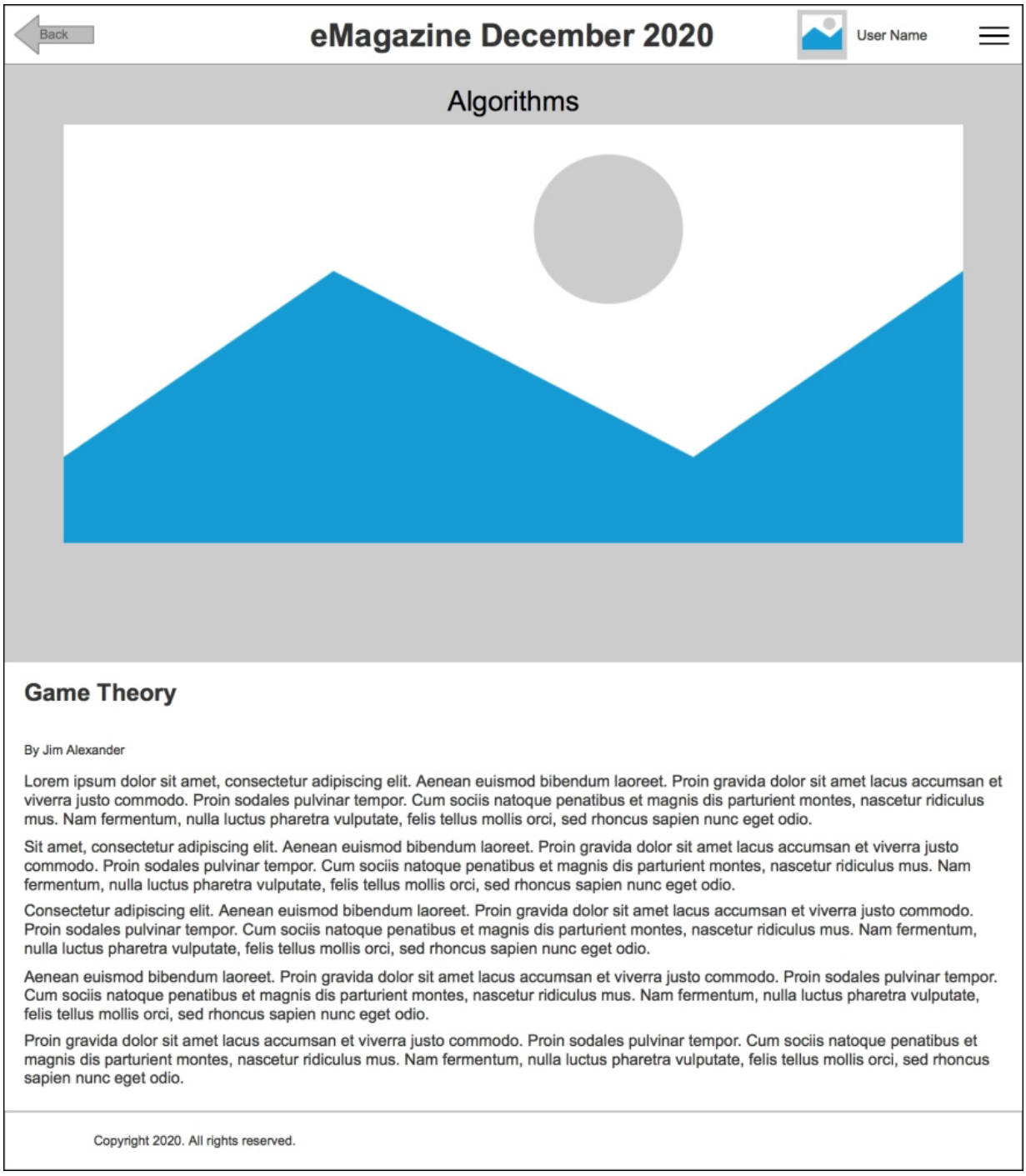
    1. Under **Click to add actions**, scroll to the **Widgets** dropdown menu and click on **Set value**. Under **Configure actions**, click on the checkbox next to **DetailTitle**. Under **Set variable to**, click on the first dropdown and then on **Value**. In the text field, enter `[[Title2]]`.
    2. Under **Configure actions**, click on the checkbox next to **DetailAuthor**. Under **Set variable to**, click on the first dropdown and then on **value**. In the text field, enter `[[Author2]]`.

16. Create the second action. To open the **Article Detail** page in the Current Window, under **Click to add actions**, in the **Links** dropdown menu under the **Open Link** dropdown menu, click on **Current Window**. Under **Configure actions**, click on **Article Detail**. Click on **OK**.
17. With **Article2Image** at coordinates (438,780) still selected, right-click on **Case 1** under the **OnClick** event and select **Copy**.
18. Click on **Article2Title** at coordinates (438,1080). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, right-click on **OnClick**, and select **Paste**.
19. Click on **Article2ReadArrow** at coordinates (920,1132). In the **Widget Interactions and Notes** pane, click on the **Interactions** tab, right-click on **OnClick**, and select **Paste**.

With the design and interactions completed for the **Topic Overview** page, we are now ready to design the **Article Detail** page.

Designing the Article Detail page

Once completed, the **Article Detail** page will look as follows:



The **Article Detail** page will leverage the global header and footer and contain the following widgets:

Widget Name	Widget Type
BackArrow	Shape
HeroImage	Image
ArticleDetailTitle	Heading 2
ArticleDetailName1	Label
ArticleParagraph1 to ArticleParagraph5	Paragraph

To design the **Article Detail** page for our eMagazine, perform the following steps:

1. In the sitemap, double-click on the **Article Detail** page to open it in the design area. Select the **Base** Adaptive View.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. From the **Masters** pane, drag the **Footer** master and drop it at coordinates (0,1110).
4. From the **Widgets** pane, drag the **Rectangle widget** and drop it at coordinates (10,10). In the toolbar, change **w:** `80` and **h:** `40`. Type `Back`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `BackArrow`. Right-click on the **Rectangle widget**, mouse over **Select Shape**, and select **Arrow Left**.
5. Repeat step 4 using the following table (* if applicable):

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
 Image	(0,60)	Topic (Note: Font 28; Padding T: 20)	1024	600	HeroImage
H2 Heading 2	(20,676)	Article Title	300	28	ArticleDetailTitle
Label	(20,740)	By Author Name	97	15	ArticleDetailName1
Paragraph	(20,770)	(Note: Adjust paragraph content to fit.)	990	54	ArticleParagraph1
Paragraph	(20,836)	(Note: Adjust paragraph content to fit.)	990	54	ArticleParagraph2
Paragraph	(20,900)	(Note: Adjust paragraph content to fit.)	990	54	ArticleParagraph3

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
 Paragraph	(20,966)	(Note: Adjust paragraph content to fit.)	990	54	ArticleParagraph4
 Paragraph	(20,1030)	(Note: Adjust Paragraph content to fit.)	990	54	ArticleParagraph5

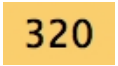
With the Base view designed, we are now ready to update the design for each adaptive view for the 768 and 320 views.

To update the 768 view, click on the 768 button 

that is located in the upper-left corner of the design area. With the Topic Overview page still open in the design area, perform the following steps:

1. In the design area, click on the Footer master. In the toolbar, change **x:** 0 and **y:** 1030.
2. Reposition and resize widgets to complete the 768 view using the following table (* if applicable):

Named Widget (In the Widget Interactions and Notes pane)	Base View Coordinates	New Coordinates*	New Width* (w:)	New Height* (h:)
HeroImage	(0,60)		768	450
ArticleDetailTitle	(20,676)	(20,530)		
ArticleDetailName1	(20,740)	(20,594)		
ArticleParagraph1	(20,770)	(20,628)	740	72
ArticleParagraph2	(20,836)	(20,710)	740	72
ArticleParagraph3	(20,900)	(20,790)	740	72
ArticleParagraph4	(20,966)	(20,870)	740	72
ArticleParagraph5	(20,1030)	(20,956)	740	54

To update the 320 view, click on the 320 button 

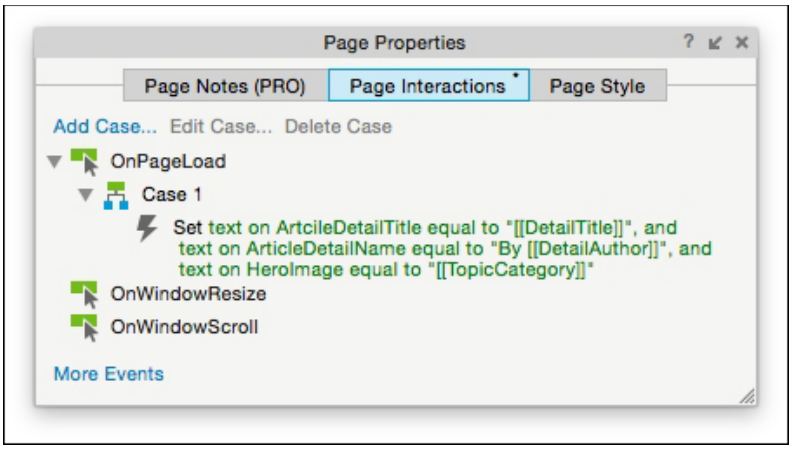
that is located in the upper-left corner of the design area. With the **Topic Overview** page still open in the design area, perform the following steps:

1. In the design area, click on the **Footer** master. In the toolbar change **x:** 0 and **y:** 1280.
2. Reposition and resize widgets to complete the 320 view using the following table (* if applicable):

Named Widget (In the Widget Interactions and Notes pane)	768 View Coordinates	New Coordinates*	New Width* (w:)	New Height* (h:)
BackArrow	(5,16)		50	30
HeroImage	(0,60)		320	188
ArticleDetailTitle	(20,530)	(10,270)	300	
ArticleDetailName1	(20,594)	(10,334)		
ArticleParagraph1	(20,628)	(10,360)	300	180
ArticleParagraph2	(20,710)	(10,558)	300	180
ArticleParagraph3	(20,790)	(10,750)	300	180
ArticleParagraph4	(20,870)	(10,928)	300	162
ArticleParagraph5	(20,956)	(10,1106)	300	144

Defining the OnPageLoad interaction

OnPageLoad of the **Article Detail** page, we want to set text on our **HeroImage**, **ArticleDetailTitle**, and **ArticleDetailName** widgets. Once completed, the **Page Properties** pane's **Page Interactions** tab will look as follows:



Under the design area in the **Page Properties** pane, click on the **Page Interactions** tab. Double-click on the **OnPageLoad** event. A **Case Editor** dialog box will open. In the **Case Editor** dialog, create the action. To set text on the widgets, follow these steps:

1. Under **Click to add actions**, scroll to the **Widgets** dropdown menu and click on **Set Text**.
2. Under **Configure actions**, click on the checkbox next to **HeroImage**.
3. Under **Set text to**, click on the first dropdown and then on **Value**. In the text field, enter `[[TopicCategory]]`.
4. Repeat steps 2 and 3 using the following table for to **Set text** for **widgets** and for **Value**:

Variables to set	**Value**
ArticleDetailTitle	**[[DetailTitle]]**
ArticleDetailName	**[[DetailAuthor]]**

5. Click on **OK**.

Defining the OnClick interactions

Next, we will define the individual **OnClick** interactions for our **Article Detail** page. With the **Article Detail** page open in the design area, select the **Base**

Adaptive View. Click on **BackArrow** at coordinates (10,10). To open **Topic Overview** in the Current Window, under **Click to add actions**, in the **Links** dropdown menu under the **Open Link** dropdown menu, click on **Current Window**. Under **Configure actions**, click on **Topic Overview**. Click on **OK**.

Congratulations! We have finished our eMagazine prototype. We can now publish to AxShare and view on our mobile devices.

Summary

We created eMagazine optimized for iPad and iPhone devices by using adaptive views. We created our Base view with 1024 width to cover the iPad landscape orientation. We also created a view at 768 width for iPad portrait and a view at 320 width for iPhone. The eMagazine comprised of **eMagazine Home**, **Topic Overview**, and **Article Detail** pages. We leveraged a single repeater widget to create the content for the eMagazine home page and used several global variables to track the user's category and article selections, enabling us to dynamically update content on the **Topic Overview** and **Article Detail** pages.

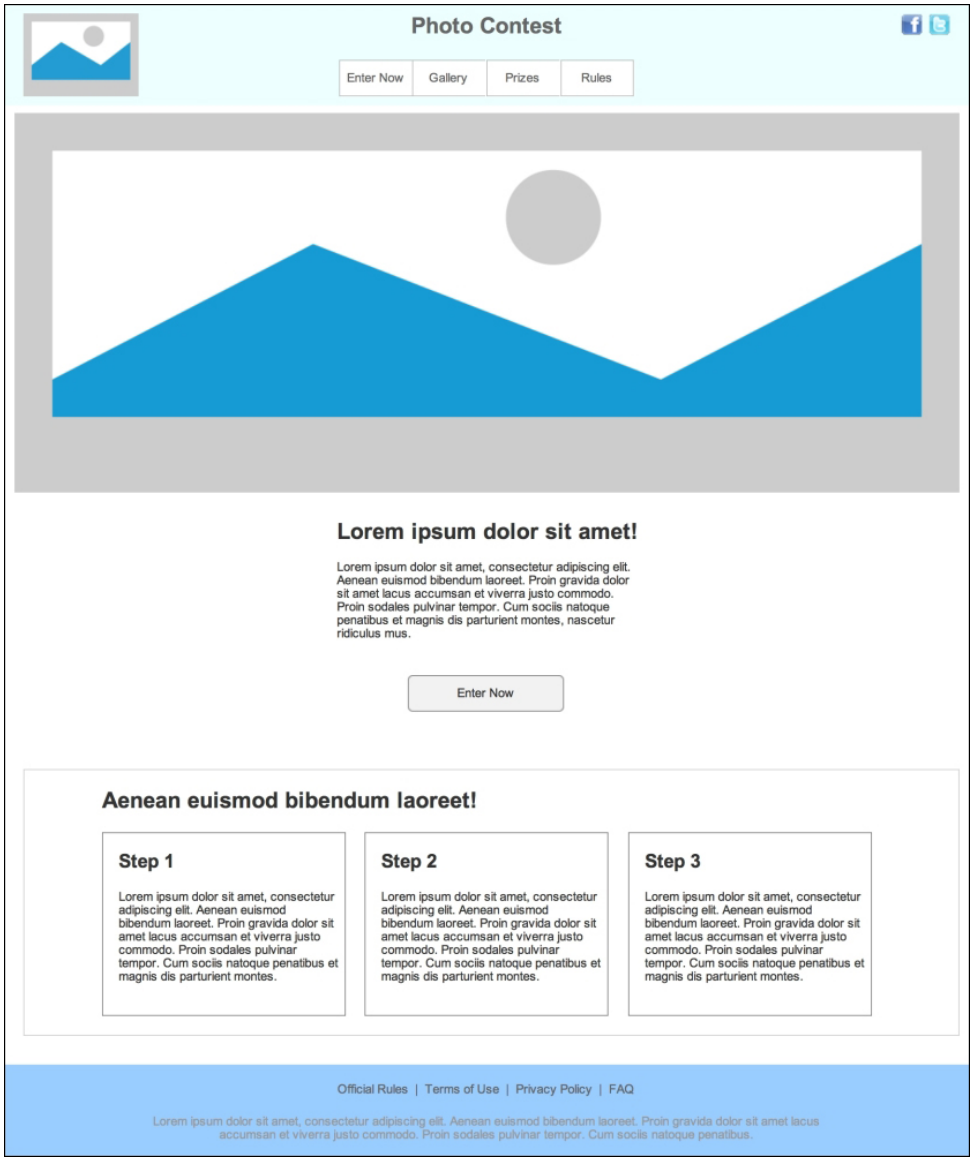
In the next chapter, we will explore building a sample photo contest, also optimized for tablet and mobile interactions.

Chapter 8. Creating a Photo Contest

Social photo sharing sites such as Facebook, Instagram, and Pinterest are extremely popular today. As a result, marketers often leverage photo contests to increase awareness of a product or service. We will continue our exploration of **Adaptive Views** by creating a photo contest tailored for the iPad and iPhone. To accomplish this, once again, we will leverage two Adaptive Views:

- Portrait tablet (768 x any and below)
- Portrait phone (320 x any and below)

As before, we will design our base view targeting a landscape tablet view of 1,024 pixels (that is, optimized to be backwards compatible with the original iPad). Our home page for our photo contest will look as follows:



In this chapter, we will cover the following topics:

- Designing a photo contest
- Creating Photo Contest interactions

Designing a photo contest

In our photo contest, we will have eight pages as follows:

- **Home**
- **Enter**
- **Upload Photo**
- **Entry Confirmation**
- **Gallery**
- **View Entry**
- **Prizes**
- **Rules**

We will start by renaming pages in our sitemap. We will then configure our Adaptive Views. Next, we will create a global header and footer that will be used by all pages. For the home page, we will have a hero image and copy with an **Enter Now** call to action. In the coming section, we will provide instructional copy outlining the steps to enter the contest. The Enter Now process will consist of the **Enter**, **Upload Photo**, and **Entry Confirmation** pages. The **Gallery** page will contain an optimized Repeater widget designed to nicely adapt for each of our targeted viewports. Finally, the **View Entry** page will contain an Image, Heading 2, Label, Shape, and social media icon widgets.

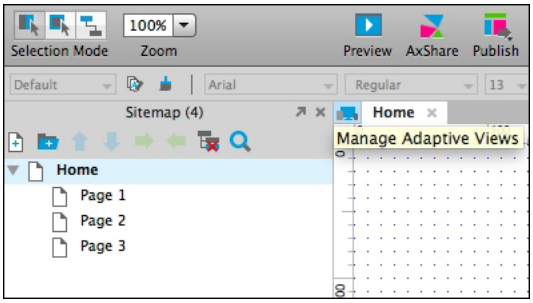
Updating the sitemap and configuring Adaptive Views

To update the sitemap and configure the Adaptive Views for our photo contest, perform the following steps:

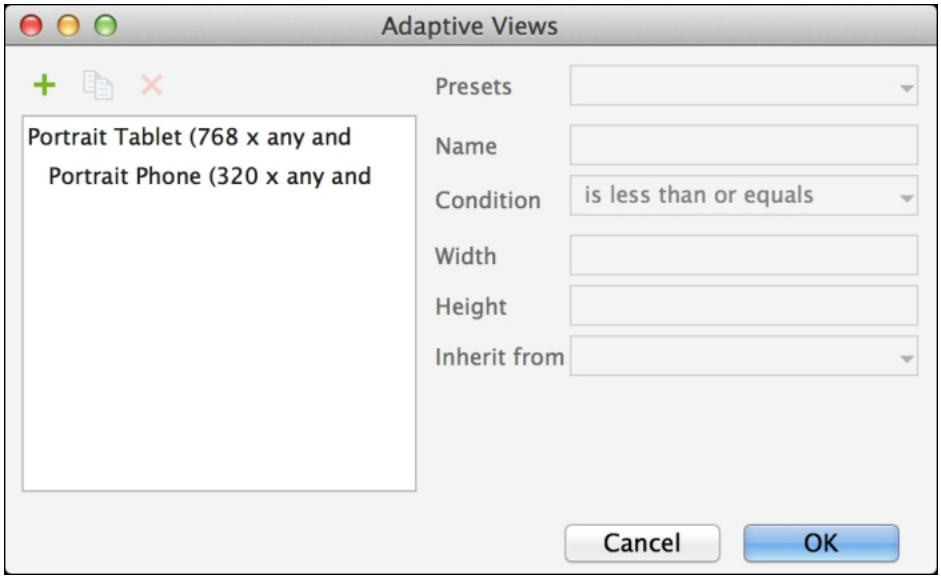
1. Open a new Axure document and organize the sitemap as follows:

    - **Home**
    - **Enter**
    - **Upload Photo**
    - **Entry Confirmation**
    - **Gallery**
    - **View Entry**
    - **Prizes**
    - **Rules**

2. To configure the Adaptive Views, click on the Manage Adaptive Views icon located in the upper-left corner of the design area as shown in the following screenshot:



3. In the **Adaptive Views** dialog box, click on the green **+** sign, and from the **Presets** drop-down menu, select **Portrait Tablet**. Click on the green **+** sign again, and from the **Presets** dropdown, menu select **Portrait Phone**. Click on **OK**. Once completed, your **Adaptive Views** dialog box should look like the following:



With our sitemap updated and the Adaptive Views configured, we will create global variables as well as the design for our **Header** and **Footer** masters.

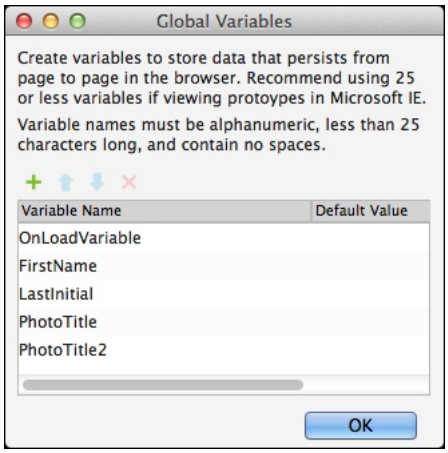
Creating global variables

We will now create global variables that will allow us share data between pages in our design. To create global variables, in the main menu click on **Project** and then **Global Variables…**. In the **Global Variables** dialog, perform the following steps:

1. Click on the green **+** sign and type `FirstName`. We will leave the **Default Value** field empty.
2. Repeat step 1 eight more times to create additional variables using the values given in the following table for the **Variable Name** field:

Variable Name
`LastInitial`
`PhotoTitle`
`PhotoTitle2`

Once completed, the **Global Variables** dialog will look like this:



3. Click on **OK**.
4. With our **global variables** created, we are now ready to complete the design and interactions for the **Header** and **Footer masters**. We will start with adding **masters** to the **Masters pane**.

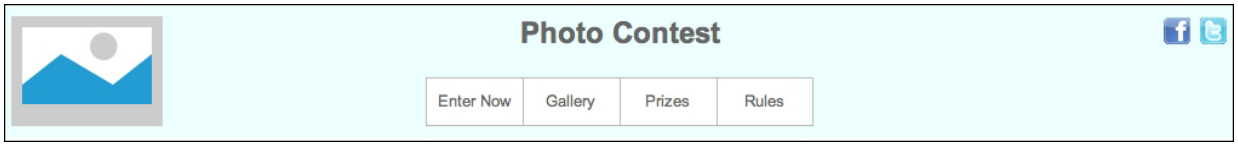
Adding masters to the Masters pane

To create the masters, perform the following steps:

1. In the **Masters** pane, click on the Add Master icon, type `Header`, and press *Enter*.
2. Next, right-click on the icon next to the **Header** master, mouse over **Drop Behavior**, and click on **Lock to Master Location**.
3. Finally, click on the Add Master icon, type `Footer`, and press *Enter*.
4. With our masters added to the **Master** pane, we are now ready to complete the design and interactions for each individual master. We will start with the **Header** master.

Designing the Header master

Our **Header** master will be used on all pages of our photo contest as identified in our sitemap. Once completed, the **Header** master will look as follows:



We will now create a solid color background for the header that will stretch to cover the width of the browser window. We will use a dynamic panel leveraging a back color with **Background Repeat** set to **Stretch to Cover**. Perform the following steps to design the **Header** master:

1. In the **Masters** pane, double-click on the icon next to the **Header** master to open it in the design area.
2. From the **Widgets** pane, drag the **Dynamic Panel** widget and place it at coordinates (0,0). In the toolbar, change **w:** to `1024` and **h:** to `110`. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `HeaderBackground`.
3. With the **HeaderBackground Dynamic Panel** still selected, click on the 768 button 

located in the upper-left corner of the design area. In the toolbar, change **w:** to `768`.
4. With the **HeaderBackground Dynamic Panel** still selected, click on the 320 button 

located in the upper-left corner of the design area. In the toolbar, change **w:** to `320` and **h:** to `127`.

5. Click on the **Base** button located in the upper-left corner of the design area. In the **Widget Manager** pane, double-click on **State1** to open it in the design area. Under the design area with the **Panel State Style** tab selected, select a **Back Color** and set **Repeat to Stretch to Cover**.

To place widgets for the **Header** master with **State1** open in the design area, select the **Base** Adaptive View and perform the following steps:

1. From the **Widgets** pane, drag the **Heading 1** widget H1 and place it at coordinates (430,10). Type `Photo Contest` in this widget. In the toolbar, change **w:** to `153` and **h:** to `28`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ContestHeading`.
2. Repeat step 2 to complete the **Base** view of the **Header** master using the following table (* If applicable):

Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
Image	(10,10)		125	90	Logo
Image	(960,10)		24	24	FacebookIcon
Image	(990,10)		24	24	TwitterIcon
Classic Menu - Horizontal	(352,60)		320	40	Menu

Note

Facebook and Twitter brand assets can be downloaded from https://www.facebookbrand.com/ and https://about.twitter.com/press/brand-assets respectively.

To complete the basic design for the Base view of the **Header** master, we need to adjust the width and height and name each of the menu items in the Menu at (352,60). To do this, perform the following steps:

1. Click on the first menu item and type `Enter Now`. In the toolbar, change **w:** to `80` and **h:** to `40`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `EnterMenuItem`.

Navigate to **Open Link** | **Links** drop-down menu under the **Click to add actions** column and click on **Current Window**. Under **Configure actions**, click on **Enter**. Click on **OK**.

2. Click on the second menu item and type `Gallery`. In the toolbar, change **w:** to `80` and **h:** to `40`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `GalleryMenuItem`. Under **Click to add actions**, in the **Links** drop-down menu under the **Open Link** drop-down menu, click on **Current Window**. Under **Configure actions**, click on **Gallery**. Click on **OK**.
3. Click on the third menu item and type `Prizes`. In the toolbar, change **w:** to `80` and **h:** to `40`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `PrizesMenuItem`. Under **Click to add actions**, in the **Links** drop-down menu under the **Open Link** drop-down menu, click on **Current Window**. Under **Configure actions**, click on **Prizes**. Click on **OK**.
4. Right-click on the third menu item and then on **Add Menu Item After**. Click on the fourth menu item and type `Rules`. In the toolbar, change **w:** to `80` and **h:** to `40`. In the **Widget Interactions and Notes** pane, click on the **Menu Item Name** field and type `RulesMenuItem`. Under **Click to add actions**, in the **Links** drop-down menu under the **Open Link** drop-down menu, click on **Current Window**. Under **Configure actions**, click on **Rules**. Click on **OK**.

To update widgets for the **Header** master in the portrait tablet view with **State1** open in the design area, click on the 768 button  located in the upper-left corner of the design area.

With the **Header** master open in the design area, reposition and resize widgets to complete the 768 view using the following table (* If applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**Base view coordinates**	**New coordinates***	**New width*** **(w:)**	**New height*** **(h:)**	**New font***
ContestHeading	(430,10)	(302,10)			
FacebookIcon	(960,10)	(710,10)			
TwitterIcon	(990,10)	(740,10)			
Menu	(352,60)	(224,60)			

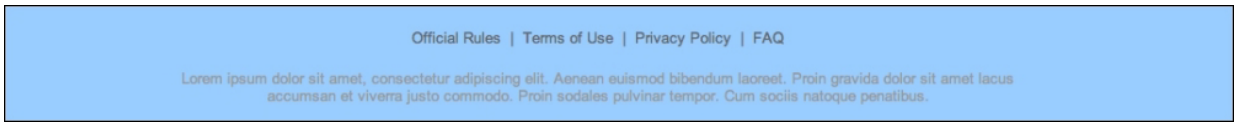
To update widgets for the **Header** master in the portrait tablet view with **State1** open in the design area, click on the 320 button  located in the upper-left corner of the design area. With the **Header** master open in the design area, reposition and resize widgets to complete the 320 view using the following table (* If applicable):

Named widget (In the Widget Interactions and Notes pane)	768 view coordinates	New coordinates*	New width* (w:)	New height* (h:)	New font*
Logo	(10,10)	(5,10)	65	60	
ContestHeading	(302,10)	(78,26)			
FacebookIcon	(710,10)	(260,10)			
TwitterIcon	(740,10)	(290,10)			
Menu	(224,60)	(0,85)			

We will now design our **Footer** master.

Designing the Footer master

Our **Footer** master will be used on all pages of our photo contest as identified in our sitemap. Once complete, the **Footer** master will look as follows:



We will now create a solid color background for the footer that will stretch to cover the width of the browser window. We will use a dynamic panel leveraging a back color with the **Background Repeat** set for **Stretch to Cover**. To do so, perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Footer** master to open it in the design area.
2. From the **Widgets** pane, drag the **Dynamic Panel** widget and place it at coordinates (0,1150). In the toolbar, change **w:** to `1024` and **h:** to `100`. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `FooterDP`.
3. With the **FooterDP Dynamic Panel** still selected, click on the 768 button 768 located in the upper-left corner of the design area. In the toolbar, change **y:** to `918`, **w:** to `768`, and **h:** to `106`.
4. With the **FooterDP Dynamic Panel** still selected, click on the 320 button 320 located in the upper-left corner of the design area. In the toolbar, change **w:** to `320` and **h:** to `152`.
5. In the **Widget Manager** pane, double-click on **State1** to open it in the design area. Under the design area with the **Panel State Style** tab selected, select a **Back Color** and set **Repeat to Stretch to Cover**.

To place widgets for the **Footer** master with **State1** open in the design area, select the **Base** Adaptive View and perform the following steps:

1. From the **Widgets** pane, drag the **Label** widget and place it at coordinates (349,20). In this widget, type `Official Rules | Terms of Use | Privacy Policy | FAQ`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `FooterLinks`.
2. From the **Widgets** pane, drag the **Label** widget and place it at coordinates (142,55). Type Greek placeholder text to be displayed on the Label widget. In the toolbar, change **w:** to `740` and **h:** to `30`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `FooterLegal`. In the **Widget Properties and Style** pane, with the **Style** tab selected, scroll to **Font** and perform the following steps:

    1. Click on the down arrow next to the text color icon . In the **#** text field in the drop-down menu, enter `999999`.
    2. Under the **Alignment + Padding** dropdown, click on the Align Center icon .

To update widgets for the **Footer** master in the portrait tablet view with **State1** open in the design area, click on the 768 button  located in the upper-left corner of the design area and reposition and resize the widgets to complete the 768 view using the following table (* If applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**Base view coordinates**	**New coordinates***	**New width*** **(w:)**	**New height*** **(h:)**
FooterLinks	(349,20)	(221,20)		
FooterLegal	(142,55)	(102,50)	`564`	`42`

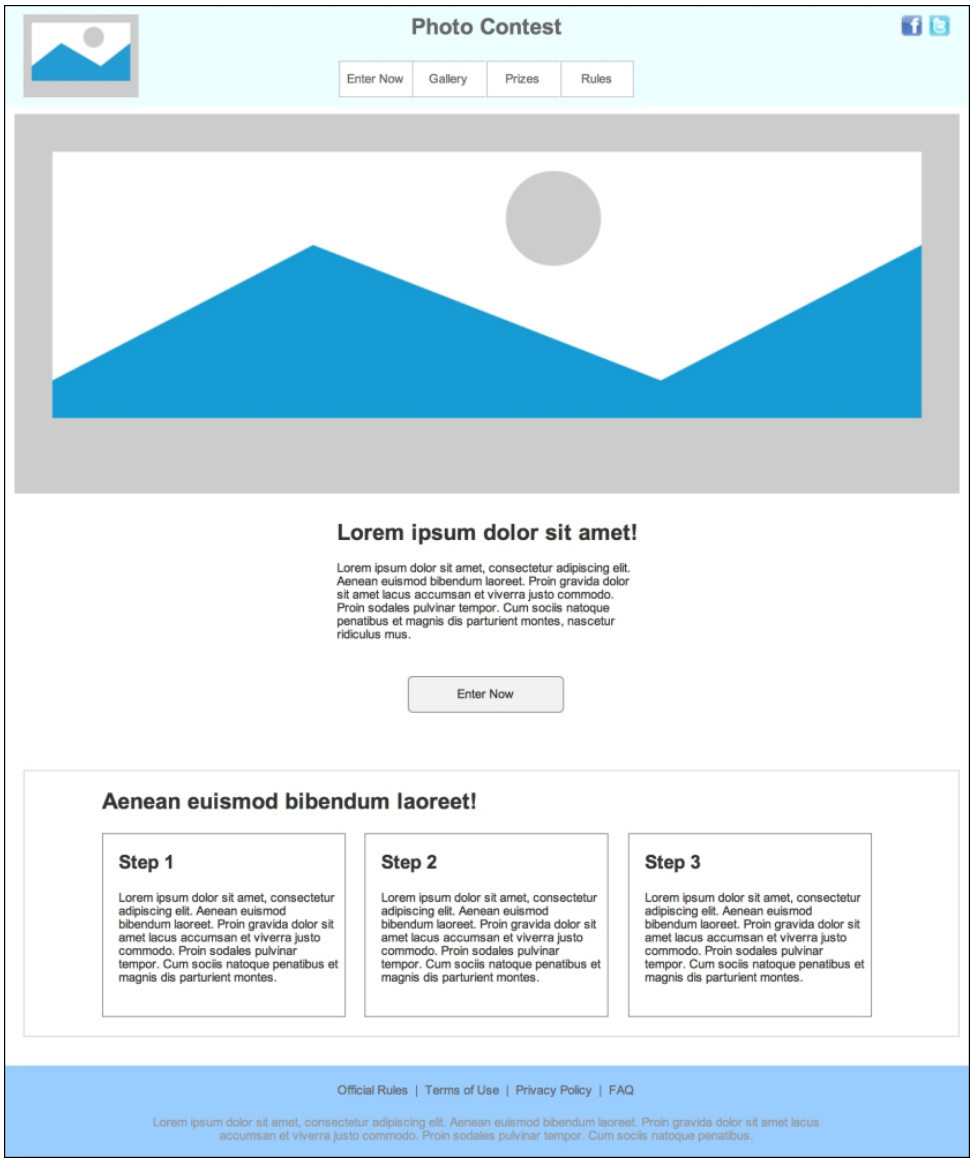
To update widgets for the **Footer** master in the portrait tablet view with **State1** open in the design area, click on the 320 button  located in the upper-left corner of the design area and reposition and resize the widgets to complete the 320 view using the following table (* If applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**768 view coordinates**	**New coordinates***	**New width*** **(w:)**	**New height*** **(h:)**	**New font***
FooterLinks	(221,20)	(15,20)	`290`	`14`	12
FooterLegal	(102,50)	(42,50)	`237`	`84`	

With the **Header** and **Footer** masters complete, we are now ready to design the home page.

Designing the Photo Contest home page

Once complete, the home page for our photo contest will look as follows:



To design the home page for our photo contest, perform the following steps:

1. In the sitemap, double-click on the **Home** page to open it in the design area. Select the **Base** Adaptive View.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. From the **Masters** pane, drag the **Footer** master and drop it at coordinates (0,1150).
4. From the **Widgets** pane, drag the **Rectangle widget** and drop it at coordinates (10,830). In the toolbar, change **w:** to `1014` and **h:** to `290`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `BackRectangle`.

> **Note**

If you choose, you can now adjust the Fill Color, Line Color, and Text Color of the Rectangle widget to blend with the other elements.

5. Repeat step 4 using the following table (* If applicable):

Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
 Image	(0,118)		1024	412	HeroImage
H2 Heading 2	(350,558)	Lorem ipsum dolor sit amet!	297	28	HeroHeadline
 Paragraph	(350,603)	(Note: Adjust paragraph content to fit.)	324	90	HeroCopy
 HTML Button	(427,728)	Enter Now (Note: To add the **OnClick** interaction, under **Click to add actions**, in the **Links** drop-down menu under the **Open Link** drop-down menu, click on **Current Window**. Under **Configure actions**, click on **Enter**. Click on **OK**.)	170	40	EnterCTAButton
H2 Heading 2	(95,850)	Aenean euismod bibendum laorett!	380	28	Heading2
 Rectangle Widget	(95,898)		264	200	Step1Rectangle
H2 Heading 2	(113,919)	Step 1 (Note: Adjust font size to **20**)	59	23	Step1Title

Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
Paragraph	(113,961)	(Note: Adjust paragraph content to fit.)	239	117	Step1Copy
Rectangle Widget	(380,898)		264	200	Step2Rectangle
H2 Heading 2	(398,919)	Step 2 (Note: Adjust font size to **20**)	59	23	Step2Title
Paragraph	(398,961)	(Note: Adjust paragraph content to fit.)	239	117	Step2Copy
Rectangle Widget	(665,898)		264	200	Step3Rectangle
H2 Heading 2	(683,919)	Step 3 (Note: Adjust font size to **20**)	59	23	Step3Title
Paragraph	(683,961)	(Note: Adjust paragraph content to fit.)	239	117	Step3Copy

With the Base view designed, we are now ready to update the design of each Adaptive View for the 768 and 320 views.

To update the 768 view, click on the 768 button 

located in the upper-left corner of the design area. With the home page still open in the design area, perform the following steps:

1. Click on the Footer master in the Design area. In the toolbar, change **x:** 0 and **y:** 1080.
2. Reposition and resize widgets to complete the 768 view using the following table (* If applicable):

Named widget (In the Widget Interactions and Notes pane)	Base view coordinates	New coordinates*	New width* (w:)	New height* (h:)	New font*
HeroImage	(0,118)	(0,115)	768	230	
HeroHeadline	(350,558)	(222,360)			
HeroCopy	(350,603)	(226,394)	315	83	**12**
EnterCTAButton	(427,728)	(324,480)	119	30	
BackRectangle	(10,830)	(0,540)	768	271	
Heading2	(95,850)	(7,552)			
Step1Rectangle	(95,898)	(7,596)	239		
Step1Title	(113,919)	(22,616)			
Step1Copy	(113,961)	(22,658)	218		
Step2Rectangle	(380,898)	(264,596)	239		
Step2Title	(398,919)	(281,616)			
Step2Copy	(398,961)	(281,658)	218		
Step3Rectangle	(665,898)	(522,596)	239		
Step3Title	(683,919)	(537,616)			
Step3Copy	(683,961)	(537,658)	218		

To update the 320 view, click on the 320 button  located in the upper-left corner of the design area. With the home page still open in the design area, perform the following steps:

1. Click on the **Footer** master in the design area. In the toolbar, change **x:** to `0` and **y:** to `985`.
2. Reposition and resize widgets to complete the 320 view using the following table (* If applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**768 view coordinates**	**New coordinates***	**New width*** **(w:)**	**New height*** **(h:)**	**New font***
HeroImage	(0,115)	(0,130)	`320`	`175`	
HeroHeadline	(222,360)	(25,311)	`270`	`23`	**20**
HeroCopy	(226,394)	(2,338)	`315`	`71`	**11**
EnterCTAButton	(324,480)	(100,414)			
BackRectangle	(0,540)	(0,462)	`320`	`493`	
Heading2	(7,552)	(7,477)	`285`	`21`	**18**
Step1Rectangle	(7,596)	(5,507)	`310`	`140`	
Step1Title	(22,616)	(26,517)			
Step1Copy	(22,658)	(26,546)	`270`	`90`	
Step2Rectangle	(264,596)	(5,657)	`310`	`140`	
Step2Title	(281,616)	(26,667)			
Step2Copy	(281,658)	(26,697)	`270`	`90`	
Step3Rectangle	(522,596)	(5,807)	`310`	`140`	
Step3Title	(537,616)	(26,817)			
Step3Copy	(537,658)	(26,847)	`270`	`90`	

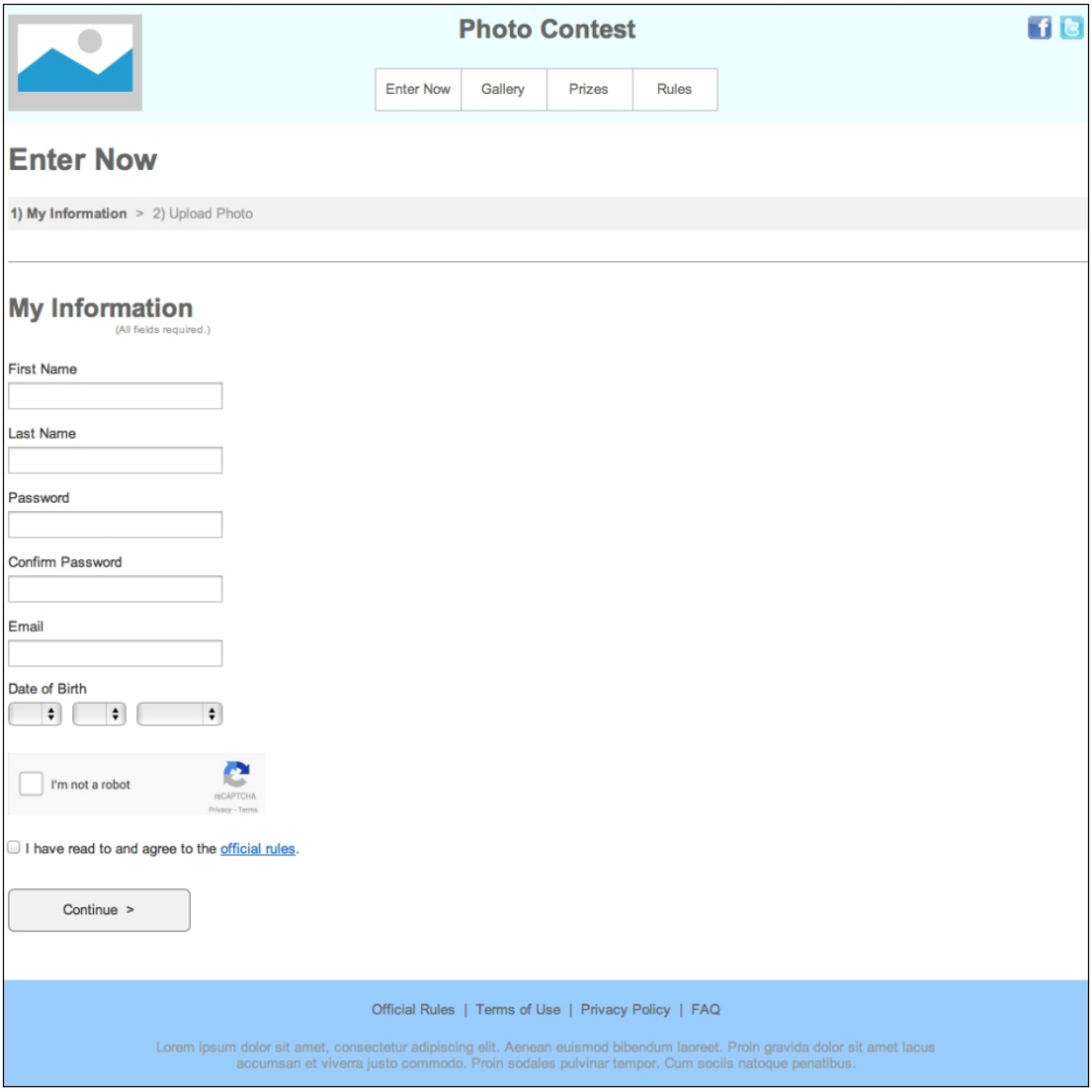
With the design and interactions completed for the **Photo Contest** home page, we are now ready to design the **Enter** page.

Creating the Enter Now flow

The Enter Now flow is comprises two pages: the **Enter** page and the **Upload Photo** page. The **Enter Now** page contains the **My Information** form. The **Upload Photo** page allows the user to upload a photo and title it. We will now design the **Enter** page.

Designing the Enter page

The **Enter** page is the first page of our **Enter Now** flow and contains the **My Information** form. Once complete, the **Enter** page will look as follows:



Note

We will leverage an optimal width (for our mobile view) across all views. This will enable us to create consistency between views as well as minimize adjustments needed for individual widget locations per view.

To design the **Enter** page for our photo contest, perform the following steps:

1. In the sitemap, double-click on the **Enter** page to open it in the design area. Select the **Base** Adaptive View.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. From the **Masters** pane, drag the **Footer** master and drop it at coordinates (0,910). Click on the 768 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `890`. Click on the 320 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `890`.
4. Click on the **Base** button located in the upper-left corner of the design area. From the **Widgets** pane, drag the **Rectangle widget** and drop it at coordinates (10,18). In the toolbar, change **w:** to `1010` and **h:** to `31`. In the widget, type `1) My Information > 2) Upload Photo`. Highlight **> 2) Upload Photo** and under **Font** change the color to `#999999`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ProgressRectangle`. Under **Alignment + Padding**, click on the left align button.
5. Repeat step 4 using the following table (* If applicable):

Widget	**Coordinates**	**Text*** **(Shown on Widget)**	**Width** **(w:)**	**Height** **(h:)**	**Name fi** **(In the V** **Interact** **Notes p**
H2 Heading 2	(10,130)	`Enter Now`			`EnterNow`
Horizontal Line	(10,235)		`1010`		`Line`
H2 Heading 2	(10,270)	`My Information` (Note: Keep font to **24**)			`MyInfoTi`
Label	(110,298)	`(All fields required.)` (Note: Keep font to **10**)			`Required`
	(10,333)	`First Name`			`FirstNam`

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name fi (In the V Interact Notes p
Label					
Text Field	(10,353)		(Note: All text fields use the defaults, that is, **w:** 200 and **h:** 25		FirstNam
Label	(10,393)	Last Name			LastName
Text Field	(10,413)				LastName
Label	(10,453)	Password			Passwor
Text Field	(10,473)				Passwor
Label	(10,513)	Confirm Password			ConfPass
Text Field	(10,533)				ConfPass
Label	(10,573)	Email			EmailLab
Text Field	(10,593)				EmailFie

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name fi (In the V Interact Notes p
Label	(10,631)	Date of Birth			DOBLabel
Droplist	(10,651)	(Note: Right-click on **Droplist** and click on **Edit List Items**. Add list items 1-12)	50	22	MonthLis
Droplist	(70,651)	(Note: Right-click on **Droplist** and click on **Edit List Items**. Add list items 1-31)	50	22	DayList
Droplist	(130,651)	(Note: Right-click on **Droplist** and click on **Edit List Items**. Add list items, for example, 1960-2000)	80	22	YearList
Image	(10,698)	(Note: a reCAPTCHA image can be found at https://www.google.com/recaptcha/intro/index.html)	240	58	reCAPTCH
Checkbox	(10,780)	I have read and agree to the official rules. (Note: Change the font color for official rules to indicate to the user that this is a clickable link)	280		OptInChe
HTML Button	(10,825)	Continue > (Note: To add the **OnClick** interaction: Under **Click to add actions**, in the **Links** dropdown menu under the **Open Link** dropdown menu and click on **Current Window**. Under **Configure actions**, click on **Upload Photo**. Click on **OK**.)	170	40	Continue

With the Base view designed, we are now ready to update the design of each adaptive view for the 768 and 320 views.

Note

Because we optimized the layout, we will only need to update two widgets for each view.

To update the 768 view, click on the 768 button  located in the upper-left corner of the design area. With the **Enter** page still open in the design area, reposition and resize widgets to complete the 768 view using the following table (* If applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**Base view coordinates**	**New width*** **(w:)**
ProgressRectangle	(10,180)	750
Line	(10,235)	750

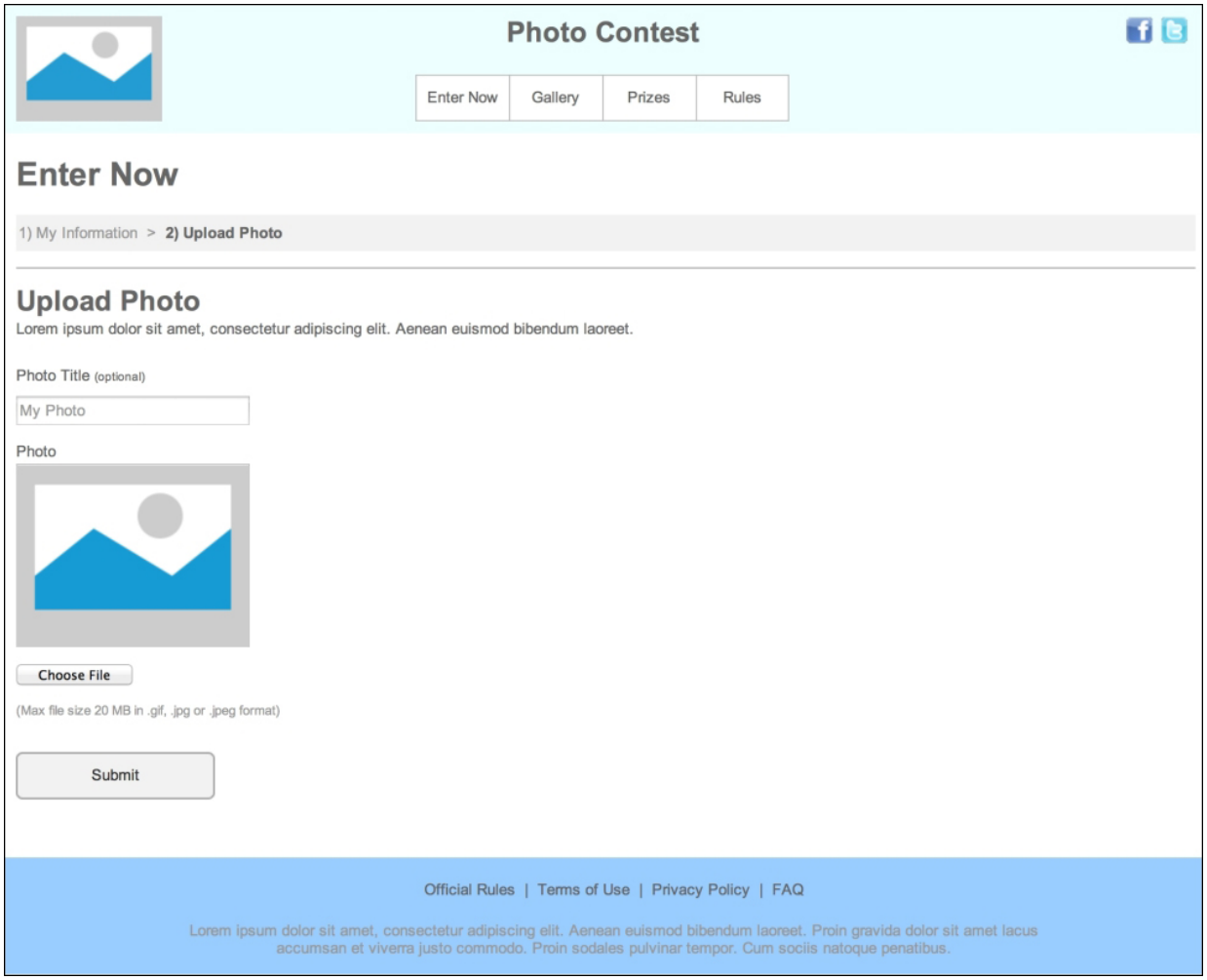
To update the 320 view, click on the 320 button  located in the upper-left corner of the design area. With the **Enter** page still open in the design area, reposition and resize widgets to complete the 320 view using the following table (* If applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**768 View Coordinates**	**New Width*** **(w:)**
ProgressRectangle	(10,180)	300
Line	(10,235)	300

With the design and interactions completed for the **Enter** page, we are now ready to design the **Upload Photo** page.

Designing the Upload Photo page

The **Upload Photo** page is the second step in our **Enter Now** flow. Once complete, the **Upload Photo** page will look like the following screenshot:



Note

We will leverage an optimal width (for our mobile view) across all views. This will enable us to create consistency between views as well as minimize adjustments needed for individual widget locations per view.

To design the **Upload Photo** page for our photo contest, perform the following steps:

1. In the sitemap, double-click on the **Upload Photo** page to open it in the design area. Select the **Base** Adaptive View.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. From the **Masters** pane, drag the **Footer** master and drop it at coordinates (0,760). Click on the 768 button 768 located in the upper-left corner of the design area. In the toolbar, change **y:** to `821`. Click on the 320 button 320 located in the upper-left corner of the design area. In the toolbar, change **y:** to `720`.
4. Click on the **Base** button located in the upper-left corner of the design area. From the **Widgets** pane, drag the **Rectangle widget** and drop it at coordinates (10,180). In the toolbar, change **w:** to `1010` and **h:** to `31`. Type `1) My Information > 2) Upload Photo` in the widget. Highlight **> 1) My Information** and under **Font**, change the color to `#999999`. In the **Widget Interactions and Notes**

pane, click on the **Shape Name** field and type `ProgressRectangle`. Under **Alignment + Padding**, click on the left align button.

5. Repeat step 4 using the following table (* If applicable):

Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
H2 Heading 2	(10,130)	`Enter Now`			`EnterNowHeading`
Horizontal Line	(10,235)		`1010`		`Line`
H2 Heading 2	(10,270)	`Upload Photo` (Note: Keep font to 24)			`UploadPhotoTitle`
Label	(10,300)	`Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.`			`InstructionalLabel`
Label	(10,340)	`Photo Title` (Note: You can add `required` or `optional` to this label. The prototype shows the text (`optional`) at font size of 10.)			`PhotoTitleLabel`
Text Field	(10,365)	(Note: You can place placeholder text by clicking on the **Properties** tab in the **Widget Properties and Style** pane and setting the **Hint Text** to **My Photo** and **Hint Style** to `#999999`.)	(Note: This text field uses the defaults of than is **w:** `200` and **h:** `25`)		`PhotoTitleField`
Label	(10,405)	`Photo`			`PhotoLabel`

Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
 Image	(10,425)		200	155	PhotoPreviewImage
 HTML Button	(10,595)	Choose file	100	25	ChooseFileButton
 Label	(10,628)	(Max file size 20 MB in .gif, .jpg, or .jpeg format) (Note: Keep font to 11 and text color to #999999)			MaxSizeLabel
 HTML Button	(10,670)	Submit (Note: To add the **OnClick** interaction, under **Click to add actions**, in the **Links** drop-down menu under the **Open Link** drop-down menu, click on **Current Window**. Under **Configure actions**, click on **Entry Confirmation**. Click on **OK**.)	170	40	SubmitButton

With the Base view designed, we are now ready to update the design of each Adaptive View for the 768 and 320 views.

To update the 768 view, click on the 768 button  located in the upper-left corner of the design area. With the **Upload Photo** page still open in the design area, reposition and resize widgets to complete the 768 view using the following table (* If applicable):

Named widget (In the Widget Interactions and Notes pane)	Base view coordinates	New coordinates*	New width* (w:)
ProgressRectangle	(10,180)		750

Named widget (In the Widget Interactions and Notes pane)	Base view coordinates	New coordinates*	New width* (w:)
Line	(10,235)		750
UploadPhotoTitle	(10,270)	(10,260)	
InstructionalLabel	(10,300)	(10,290)	
PhotoTitleLabel	(10,340)	(10,330)	
PhotoTitleField	(10,365)	(10,355)	
SubmitButton	(10,670)	(10,700)	

To update the 320 view, click on the 320 button 

located in the upper-left corner of the design area. With the **Upload Photo** page still open in the design area, reposition and resize widgets to complete the 320 view using the following table (* If applicable):

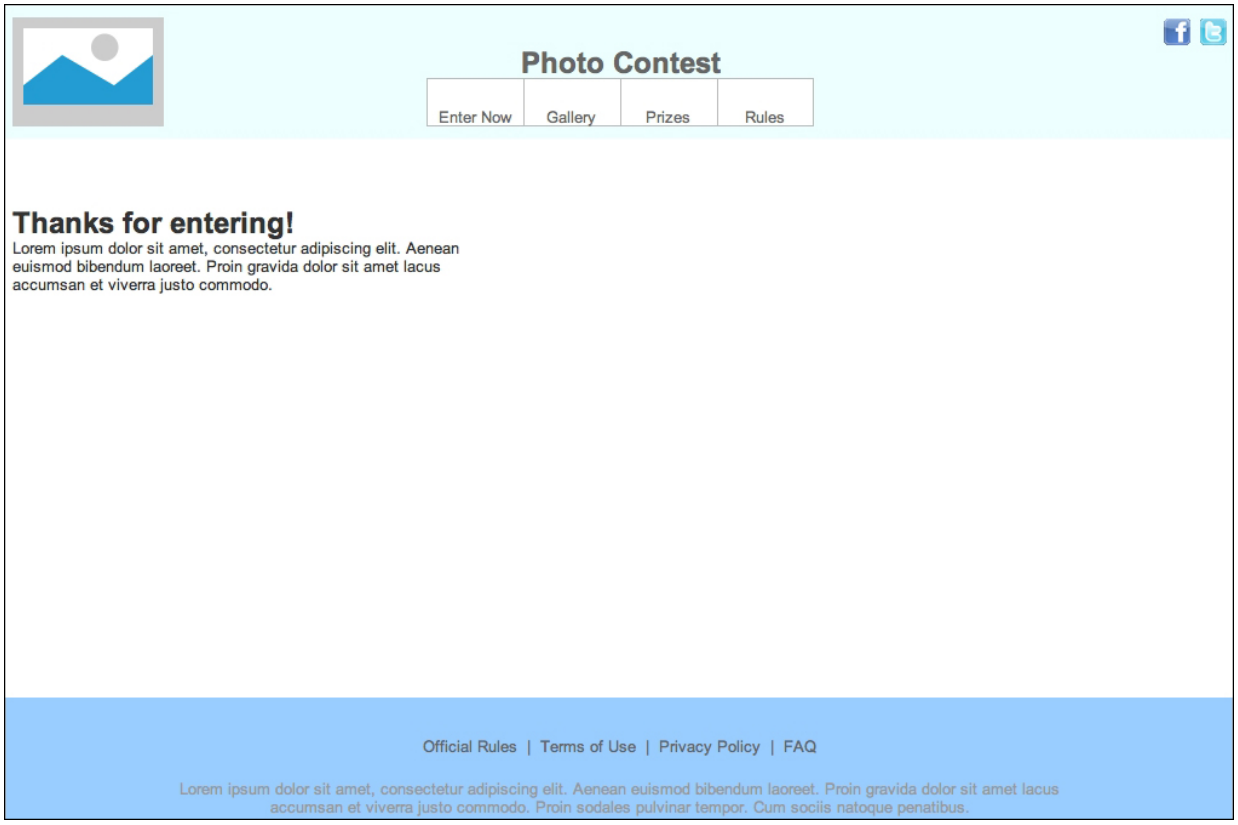
Named widget (In the Widget Interactions and Notes pane)	768 view coordinates	New coordinates*	New width* (w:)	New height* (h:)
EnterNowHeading	(10,130)	(10,140)		
ProgressRectangle	(10,180)	(10,189)	300	
Line	(10,235)	(10,245)	300	
UploadPhotoTitle	(10,260)	(10,275)		
InstructionalLabel	(10,290)	(5,313)	294	
PhotoTitleLabel	(10,330)	(10,355)		

Named widget (In the Widget Interactions and Notes pane)	768 view coordinates	New coordinates*	New width* (w:)	New height* (h:)
PhotoTitleField	(10,355)	(10,375)		
PhotoLabel	(10,405)	(10,425)		
PhotoPreviewImage	(10,425)	(10,445)	124	96
ChooseFileButton	(10,595)	(10,550)		
MaxSizeLabel	(10,628)	(10,585)		
SubmitButton	(10,700)	(10,630)		

With the design and interactions completed for the **Upload Photo** page, we are now ready to design the **Entry Confirmation** page.

Designing the Entry Confirmation page

Once complete, the **Entry Confirmation** page will look as follows:



To design the **Entry Confirmation** page for our photo contest, perform the following steps:

1. In the sitemap, double-click on the **Entry Confirmation** page to open it in the design area. Select the **Base** Adaptive View.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. From the **Masters** pane, drag the **Footer** master and drop it at coordinates (0,571). Click on the 768 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `821`. Click on the 320 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `328`.
4. Click on the **Base** button located in the upper-left corner of the design area. From the **Widgets** pane, drag and drop the **Heading 2** widget at coordinates (10,142). In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `EntryConfHeading`.
5. From the **Widgets** pane, drag and drop the **Paragraph widget** at coordinates (10,180). In the toolbar, change **w:** to `380` and **h:** to `45`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `EntryConfCopy`. Adjust the copy shown on the widget to three lines of text.

With the Base view designed, we are now ready to update the design of each Adaptive View for the 768 and 320 views. After review, we find that the widgets do not need to be updated for the 768 view. We are now ready to update the 320 view.

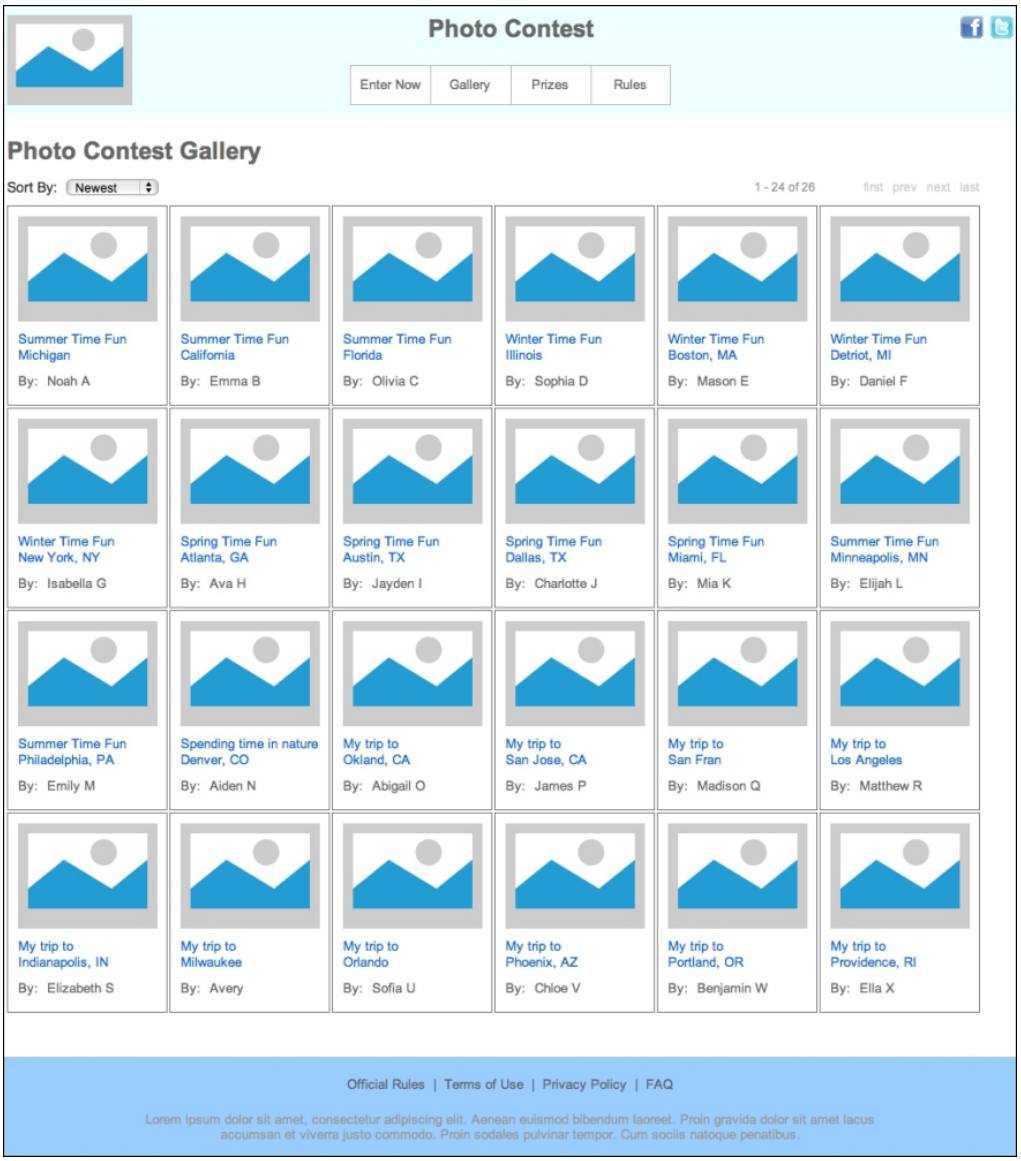
To update the 320 view, click on the 320 button  located in the upper-left corner of the design area. With the **Entry Confirmation** page still open in the design area, reposition and resize widgets to complete the 320 view using the following table (* If applicable):

Named widget (In the Widget Interactions and Notes pane)	768 view coordinates	New coordinates*	New width* (w:)	New height* (h:)	New font*
EntryConfHeading	(10,142)	(5,142)			
EntryConfCopy	(10,180)	(5,180)	294	60	

With the design complete for the **Entry Confirmation** page, we are now ready to design the **Gallery** page.

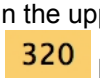
Designing the Gallery page

The content of the **Gallery** page is provided by an adaptive Repeater widget optimized for each Adaptive View. Once complete, the **Gallery** page will look like this:



To design the **Gallery** page for our photo contest, perform the following steps:

1. In the sitemap, double-click on the **Gallery** page to open it in the design area; then, select the **Base** Adaptive View.
2. From the **Masters** pane, drag and drop the **Header** master anywhere in the design area.
3. From the **Masters** pane, drag and drop the **Footer** master at coordinates (0,1050). Click on the 768 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `1864`.

   Click on the 320 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `2640`.
4. Click on the **Base** button located in the upper-left corner of the design area. From the **Widgets** pane, drag the **Heading 2** widget and drop it at coordinates (10,132). Type `Photo Contest Gallery` in this widget. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `GalleryPageTitle`.
5. Repeat step 4 using the following table (* If applicable):

Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
Label	(10,174)	`Sort By:` (Note: Keep font to 14)	51	16	SortByLabel
Droplist	(69,174)	(Note: Right-click on **Droplist** and click on **Edit List Items**. Add list items: **Newest**, **Oldest**, and **Most Viewed**. Click on the checkbox next to **Newest** to select as default.)	92	16	SortDropList
Label	(755,175)	`1-24 of 26` (Note: Set text color to `#999999`)	62	14	Count
Label	(864,175)	`First` (Note: Set text color to `#999999`)	20	14	FirstLink
Label	(893,175)	`prev` (Note: Set text color to `#999999`)	25	14	PreviousLink
Label	(927,175)	`next` (Note: Set text color to `#999999`)	24	14	NextLink
Label	(960,175)	`last` (Note: Text color - #999999)	20	14	LastLink

With the Base view designed, we are now ready to update the design of each Adaptive View for the 768 and 320 views.

Note

If you choose to add additional interactions, such as updating the **Count** label based on a different number of items shown per view, be aware that text is not adaptive. Any change you make in one

view will persist across all views.

The solution for this is to Unplace from view the **Count** label in each child view and create a new uniquely named Count (for example, Count_Desktop, Count_Tablet, and Count_Mobile). In this way, you can update each individual Count widget on page load and be certain that the proper text is shown on the Label widget.

To update the 768 view, click on the 768 button 

located in the upper-left corner of the design area. With the **Gallery** page still open in the design area, reposition and resize widgets to complete the 768 view using the following table (* If applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**Base view coordinates**	**New coordinates***
SortByLabel	(10,174)	(9,172)
SortDropList	(69,174)	(68,172)
Count	(755,175)	(508,173)
FirstLink	(864,175)	(638,173)
PreviousLink	(893,175)	(667,173)
NextLink	(927,175)	(701,173)
LastLink	(960,175)	(734,173)

To update the 320 view, click on the 320 button 

located in the upper-left corner of the design area. With the **Gallery** page still open in the design area, reposition and resize widgets to complete the 320 view using the following table (* If applicable):

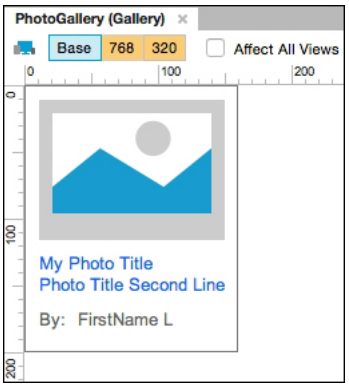
Named widget **(In the Widget Interactions and Notes pane)**	**768 view coordinates**	**New coordinates***
GalleryPageTitle	(10,135)	(5,135)

Named widget (In the Widget Interactions and Notes pane)	768 view coordinates	New coordinates*
SortByLabel	(9,172)	(5,182)
SortDropList	(68,172)	(64,182)
Count	(508,173)	(5,220)
FirstLink	(638,173)	(205,220)
PreviousLink	(667,173)	(234,220)
NextLink	(701,173)	(268,220)
LastLink	(734,173)	(301,220)

Next, we will design the photo gallery repeater.

Designing the PhotoGallery repeater

Once complete, the PhotoGallery repeater will look as follows:



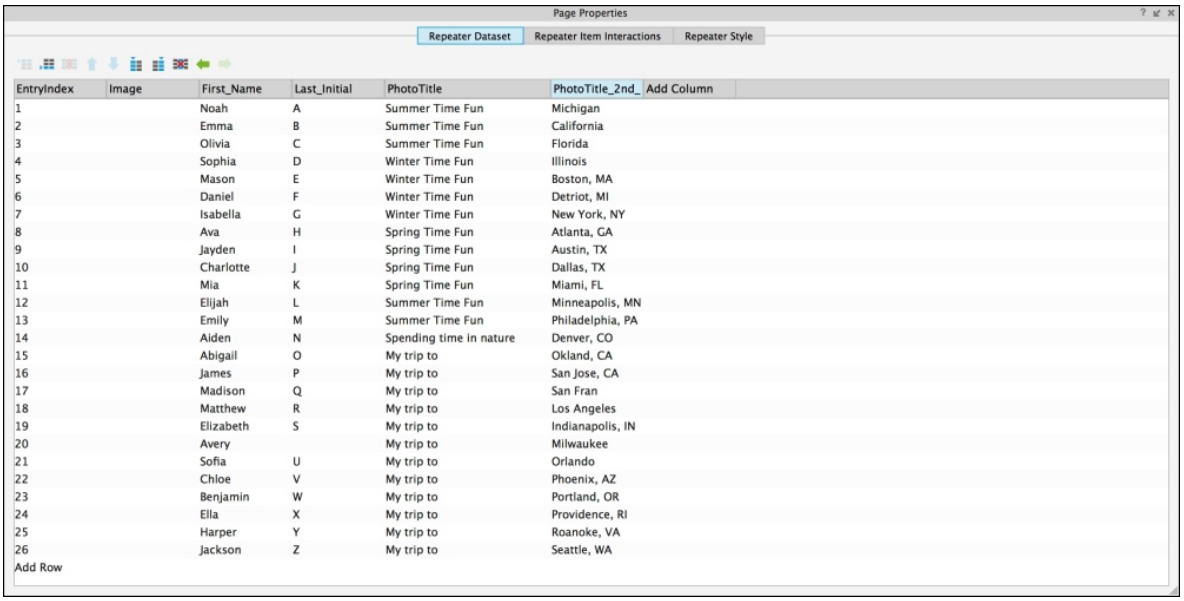
To design the PhotoGallery repeater, perform the following steps:

1. From the **Widgets** pane, drag the **Repeater** widget into the design area and place it at coordinates (10,200). In the **Widget Interactions and Notes** pane, click in the **Repeater Name** field and type `PhotoGallery`.
2. Double-click on the **PhotoGallery** Repeater widget to open in the design area.
3. Under the design area in the Repeater pane, click on the **Repeater Style** tab. To change the default **Repeater Style**, perform the following steps:

1. Under the **Layout** dropdown, click on the radio button next to **Horizontal**.
2. Select the checkbox next to **Wrap (Grid)**, and in **Items per row**, enter `6`.
3. Under the **Pagination** dropdown, select the checkbox next to **Multiple pages**. In the **Items per page** field, enter `24`. In the **Starting page** field, enter `1`.
4. Under the **Spacing** dropdown, enter `2` in the **Spacing: Row** and **Column** fields.

4. Under the design area in the Repeater pane, click on the **Repeater Dataset** tab. Use the table shown in the following screenshot to update data values:

> ## Note
>
> To rename columns or rows, double-click on the column or row and type the new column or row name. To add new columns or rows, click on **add column** or **add row**.



EntryIndex	Image	First_Name	Last_Initial	PhotoTitle	PhotoTitle_2nd_	Add Column
1		Noah	A	Summer Time Fun	Michigan	
2		Emma	B	Summer Time Fun	California	
3		Olivia	C	Summer Time Fun	Florida	
4		Sophia	D	Winter Time Fun	Illinois	
5		Mason	E	Winter Time Fun	Boston, MA	
6		Daniel	F	Winter Time Fun	Detriot, MI	
7		Isabella	G	Winter Time Fun	New York, NY	
8		Ava	H	Spring Time Fun	Atlanta, GA	
9		Jayden	I	Spring Time Fun	Austin, TX	
10		Charlotte	J	Spring Time Fun	Dallas, TX	
11		Mia	K	Spring Time Fun	Miami, FL	
12		Elijah	L	Summer Time Fun	Minneapolis, MN	
13		Emily	M	Summer Time Fun	Philadelphia, PA	
14		Aiden	N	Spending time in nature	Denver, CO	
15		Abigail	O	My trip to	Okland, CA	
16		James	P	My trip to	San Jose, CA	
17		Madison	Q	My trip to	San Fran	
18		Matthew	R	My trip to	Los Angeles	
19		Elizabeth	S	My trip to	Indianapolis, IN	
20		Avery		My trip to	Milwaukee	
21		Sofia	U	My trip to	Orlando	
22		Chloe	V	My trip to	Phoenix, AZ	
23		Benjamin	W	My trip to	Portland, OR	
24		Ella	X	My trip to	Providence, RI	
25		Harper	Y	My trip to	Roanoke, VA	
26		Jackson	Z	My trip to	Seattle, WA	

Add Row

We will now resize the Rectangle widget on the Repeater Item and place additional widgets. For this, perform the following steps:

1. With the Repeater still open in the design area, click on the Rectangle widget at coordinates (0,0). In the toolbar, change **w:** to `160` and **h:** to `200`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Background`.
2. From the **Widgets** pane, drag the **Image** widget and place it at coordinates (11,11). In the toolbar, change **w:** to `139` and **h:** to `105`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Image`.
3. From the **Widgets** pane, drag the **Label** widget and place it at coordinates (11,126). Type `My Photo Title`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `PhotoTitleLabel`.
4. From the **Widgets** pane, drag the **Label** widget and place it at coordinates (11,142). Type `Photo Title Second Line`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `PhotoTitleLabel2`.
5. From the **Widgets** pane, drag the **Label** widget and place it at coordinates (11,168). In the toolbar, change **w:** to `25` and **h:** to `15`. Type `By:`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ByLabel`.
6. From the **Widgets** pane, drag the **Label** widget and place it at coordinates (40,168). Type `FirstName L`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type

Name.

With the Base view designed for the **PhotoGallery** repeater, we are now ready to update the design of each adaptive view for the 768 and 320 views.

Updating the PhotoGallery repeater for the 768 and 320 views

To update the 768 view, click on the 768 button 

located in the upper-left corner of the design area. With the **PhotoGallery** Repeater Item still open in the design area, reposition and resize widgets to complete the 768 view using the following table (* If applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**Base view coordinates**	**New coordinates***	**Width*** **(w:)**	**Height*** **(h:)**
Background	(0,0)		184	230
Image	(11,11)	(13,13)	160	121
PhotoTitleLabel	(11,126)	(13,145)		
PhotoTitleLabel2	(11,142)	(13,163)		
ByLabel	(11,168)	(13,193)		
Name	(40,168)	(46,193)		

Under the design area in the Repeater pane, click on the **Repeater Style** tab and perform the following steps:

1. Under the **Layout** dropdown, click on the radio button next to **Horizontal**.
2. Select the checkbox next to **Wrap (Grid)**, and in **Items per row**, enter 4.
3. Under the **Pagination** dropdown, select the checkbox next to **Multiple pages**. In the **Items per page** field, enter 24. In the **Starting page** field, enter 1.
4. Under the **Spacing** dropdown, enter 2 in the **Spacing: Row** field and enter 4 in the **Column** field.

To update the 320 view, click on the 320 button 

located in the upper-left corner of the design area. With the **PhotoGallery** Repeater Item still open in the design area, perform the following steps:

Named widget (In the Widget Interactions and Notes pane)	768 view coordinates	New coordinates*	Width* (w:)	Height* (h:)
Background	(0,0)		155	195
Image	(13,13)	(5,6)	146	111
PhotoTitleLabel	(13,145)	(5,128)		
PhotoTitleLabel2	(13,163)	(5,144)		
ByLabel	(13,193)	(5,172)		
Name	(46,193)	(36,172)		

Under the design area in the Repeater pane, click on the **Repeater Style** tab. To change the default repeater style, perform the following steps:

1. Under the **Layout** dropdown, click on the radio button next to **Horizontal**.
2. Select the checkbox next to **Wrap (Grid)**, and in **Items per row**, enter `2`.
3. Under the **Pagination** dropdown, select the checkbox next to **Multiple pages**. In the **Items per page** field, enter `24`. In the **Starting page** field, enter `1`.
4. Under the **Spacing** dropdown, enter `2` in the **Spacing: Row** field and enter `4` in **Column** field.

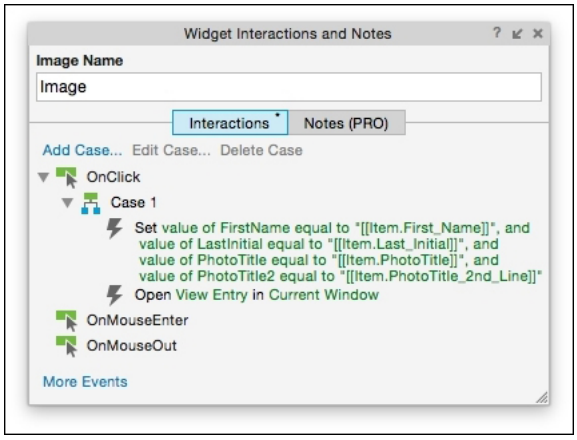
With the layout updated for the **PhotoGallery** repeater, we are now ready to define the interactions.

Defining PhotoGallery repeater interactions

Since we have already defined all of our global variables, we can now complete the interactions for our Repeater Item. We will define the **OnClick** event for the **Image** and **PhotoTitleLabel** widgets. When the user clicks on either of these widgets, the following occurs:

- The values of **FirstName**, **LastInitial**, **PhotoTitle**, and **PhotoTitle2** are set
- The **View Entry** page will open in current window

Once defined, the **Widget Interactions and Notes** pane for the Image widget will look as follows:



With the Repeater Item still open in the design area and the **Base** Adaptive View selected, in the **Widget Interactions and Notes** pane, click on the Image widget at coordinates (11,11). In the **Interactions** tab, double-click on the **OnClick** event. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

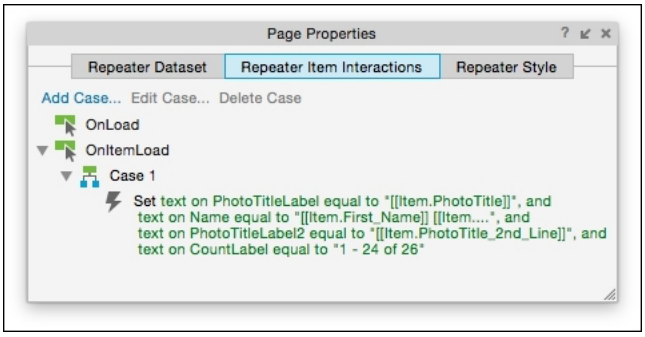
1. Create the first action. Follow these steps to set variable value for key global variables:
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, select the checkbox next to **FirstName**.
    3. Under **Set variable to**, click on the first dropdown and then on **value**. In the text field, enter `[[Item.First_Name]]`.
    4. Repeat steps 1-3 using the values given in the following table for **Variables to set** and **Value**:

Variables to set **(In the Case Editor Configure Actions section)**	**Value**
LastInitial	`[[Item.Last_Initial]]`
PhotoTitle	`[[Item.PhotoTitle]]`
PhotoTitle2	`[[Item. PhotoTitle_2nd_Line]]`

2. Create the second action. To open view entry in the current window; under **Click to add actions**, in the **Links** drop-down menu under the **Open Link** drop-down menu, click on **Current Window**. Under **Configure actions**, click on **View Entry**. Click on **OK**.
3. In the **Widget Interactions and Notes** pane, with the **Interactions** tab selected, under the **OnClick** event, right-click on **Case 1** and click on **Copy**.
4. Click on **PhotoTitleLabel** at coordinates (11,126). In the **Widget Interactions and Notes** pane, with the **Interactions** tab selected, right-click on the **OnClick** event and click on **Paste**.

To complete interactions for the Repeater, we also need to update the **OnItemLoad** event.

Once defined, the **PhotoGallery** repeater's **OnItemLoad** event will look as follows:



Under the design area in the Repeater pane, click on the **Repeater Item Interactions** tab. Double-click on the **OnItemLoad** event. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

1. Create the first action. Follow these steps to set text for widgets on the repeater item:
2. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
3. Under **Configure actions**, select the checkbox next to **PhotoTitleLabel**.
4. Under **Set text to**, click on the first dropdown and then on **value**. In the text field, enter `[[Item.PhotoTitle]]`.
5. Repeat steps 2-4 using the values given in the following table for **Widgets to set** and **Text**:

Widgets to set **(In the Case Editor Configure Actions section)**	**Text**
Name	`[[Item.First_Name]] [[Item.Last_Initial]]`
PhotoTitleLabel2	`[[Item.PhotoTitle_2nd_Line]]`
CountLabel	`1 - 24 of 26`

6. Click on **OK**.

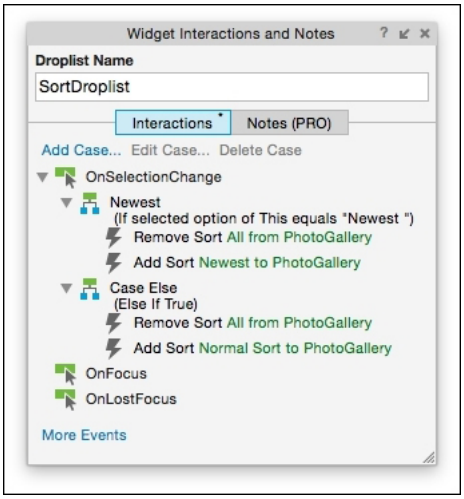
With the interactions completed for the repeater, we are now ready to define interactions for sorting and pagination for the **Gallery** page.

Defining sorting and pagination interactions

Sorting is controlled by the **OnSelectionChange** event for the **SortDroplist** widget. Pagination controls are enabled by **OnClick** events on the **FirstLink**, **PrevLink**, **NextLink**, and **LastLink** widgets. We will first enable interactions for **SortDroplist**.

Enabling sorting for SortDroplist

Once all cases for **SortDroplist** have been defined, the **Widget Interactions and Notes** pane will look as follows:



To enable interactions for the **SortDroplist** widget, do the following:

1. In the sitemap, double-click on the **Gallery** page to open it in the design area and select the **Base** Adaptive View.
2. Click on the **SortDroplist** widget at coordinates at (69,174). Double-click on the **OnSelectionChange** event. A **Case Editor** dialog box will open.
3. In the **Case Description** field within the **Case Editor** dialog, type `Newest`.
4. Now, click on the **Add Condition** button to add the condition.
5. In the outlined condition box within the **Condition Builder** dialog box, perform the following steps:

    1. In the first dropdown, select **selected option of**.
    2. In the second dropdown, select **This**.
    3. In the third dropdown, select **equals**.
    4. In the fourth dropdown, select **option**.
    5. In the fifth dropdown, select **Newest**.

6. Create the action, to remove sort all, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Repeaters** drop-down menu. Click on **Remove Sort**.
    2. Under **Configure actions**, in the **Select the repeaters to remove sorting** section, select the checkbox next to **PhotoGallery**. Then, select the checkbox next to **Remove all sorting**.

7. Create the second action, to add sort newest to the repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Repeaters** drop-down menu and click **Add Sort**.
    2. Under **Configure actions**, in the section **Select the repeaters to add sorting**, click on the checkbox next to **PhotoGallery**.
    3. In the **Name:** text field type `Newest`.
    4. In the **Property:** dropdown, select **EntryIndex**.
    5. In the **Sort as:** dropdown, select **Number**.
    6. In the **Order:** dropdown, select **Ascending**.
    7. Click on **OK**.

8. Create the **Case Else** case for the **OnSelectionChange** event. Double-click on the **OnSelectionChange** event. A **Case Editor** dialog box will open. In the **Case Description** field within the **Case Editor** dialog, type `Case Else`.
9. Create the action; to remove sort all, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Repeaters** drop-down menu. Click on **Remove Sort**.

    2. Under **Configure actions**, in the **Select the repeaters to remove sorting** section, select the checkbox next to **PhotoGallery**. Select the checkbox next to **Remove all sorting**.
10. Create another action, to add sort normal sort to the repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Repeaters** drop-down menu and click **Add Sort**.
    2. Under **Configure actions**, in the **Select the repeaters to add sorting** section, select the checkbox next to **PhotoGallery**.
    3. In the **Name:** text field, type `Normal Sort`.
    4. In the **Property:** dropdown, select **EntryIndex**.
    5. In the **Sort as:** dropdown, select **Number**.
    6. In the **Order:** dropdown, select **Descending**.
    7. Click on **OK**.

With the interactions defined for the **SortDroplist** widget, we are now ready to enable the pagination controls.

Enabling pagination controls

To enable the **OnClick** event for the **FirstLink** and **PreviousLink** widgets, click on **FirstLink** at coordinates (864,175). In the **Widget Interactions and Notes** pane, double-click on the **OnClick** event. A **Case Editor** dialog box will open. In the **Case Description** field within the **Case Editor** dialog, perform the following steps:

1. Create the first action, to set current page of the repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Repeaters** drop-down menu and click on **Set Current Page**.
    2. Under **Configure actions**, in the **Select the repeaters to set current page** section, select the checkbox next to **PhotoGallery**.
    3. In the **Select the page:** dropdown, click on **Value**.
    4. In the **Page #** text field, type `1`.

2. Create the second action. Follow these steps to set text for the **Count** widget:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, select the checkbox next to **Count**.
    3. Under **Set text to**, click on the first dropdown and then on **value**. In the text field, enter `1-24 of 26`.
    4. Click on **OK**.

3. In the **Widget Interactions and Notes** pane, with the **Interactions** tab selected, under the **OnClick** event, right-click on **Case 1** and click on **Copy**.
4. Click on **PreviousLink** at coordinates (893,175). In the **Widget Interactions and Notes** pane, with the **Interactions** tab selected, right-click on the **OnClick** event and click on **Paste**.

To enable the **OnClick** event for the **NextLink** and **LastLink** widgets, click on **NextLink** at coordinates (927,175). In the **Widget Interactions and Notes** pane, double-click on the **OnClick** event. A **Case Editor** dialog box will open. Within the **Case Editor** dialog in the **Case Description** field, perform the following steps:

1. Create the first action, to set current page of the repeater, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Repeaters** drop-down menu and click **Set Current Page**.
    2. Under **Configure actions**, in the **Select the repeaters to set current page** section, select the checkbox next to **PhotoGallery**.
    3. In the **Select the page:** dropdown, click on **Value**.
    4. In the **Page #** text field, type `2`.

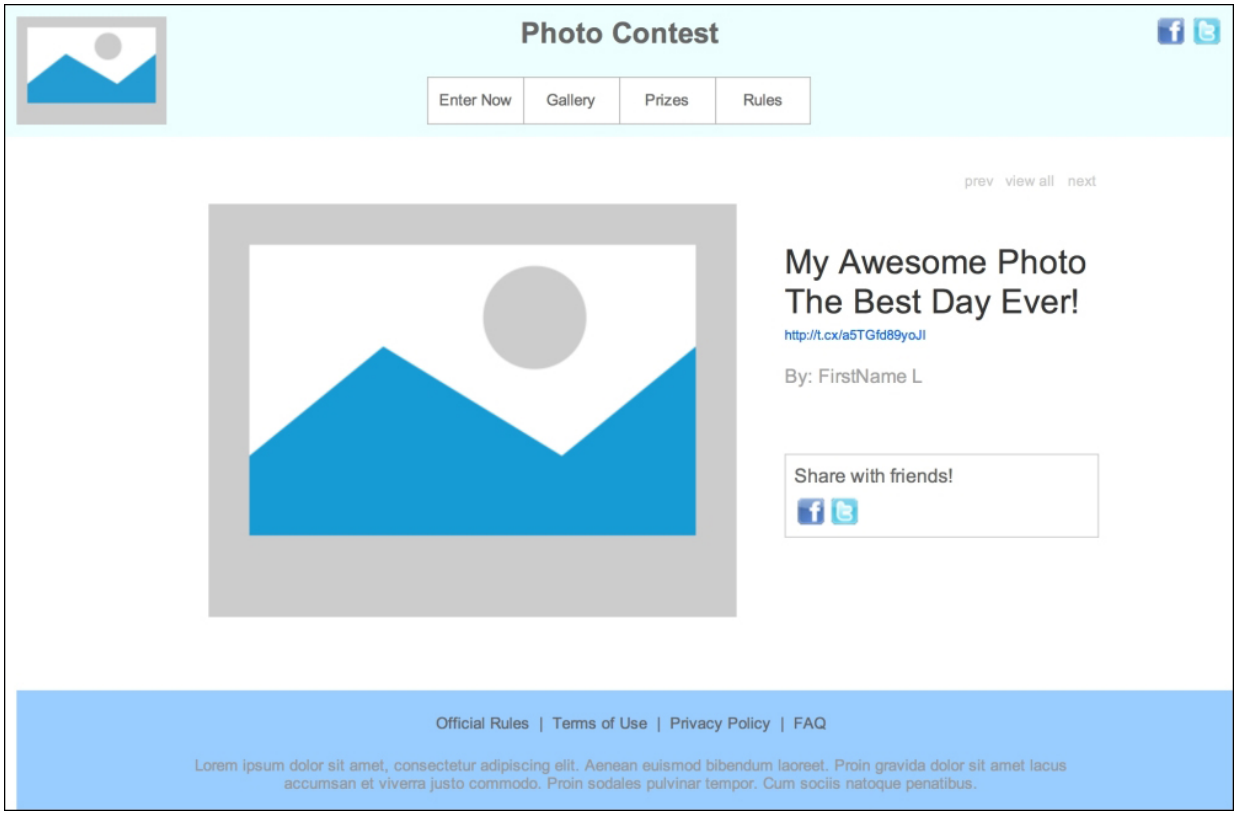
2. Create the second action. Follow these steps to set text for the count widget:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Text**.
    2. Under **Configure actions**, select the checkbox next to **Count**.
    3. Under **Set text to**, click on the first dropdown and then on **value**. In the text field, enter `25-26 of 26`.
    4. Click on **OK**.
    5. In the **Widget Interactions and Notes** pane, with the **Interactions** tab selected, under the **OnClick** event, right-click on **Case 1** and click on **Copy**.
    6. Click on the **LastLink** widget at coordinates (960,175). In the **Widget Interactions and Notes** pane, with the **Interactions** tab selected, right-click on the **OnClick** event and click on **Paste**.

With the design and interactions completed for the **Gallery** page, we are now ready to design the **View Entry** page.

Designing the View Entry page

Once complete, the **View Entry** page will look as follows:



To design the **View Entry** page for our photo contest, perform the following steps:

1. In the sitemap, double-click on the **View Entry** page to open it in the design area. Select the **Base** Adaptive view.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. From the **Masters** pane, drag the **Footer** master and drop it at coordinates (0,571). Click on the 768 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `818`. Click on the 320 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `749`.

4. Click on the **Base** button located in the upper-left corner of the design area. From the **Widgets** pane, drag the **Rectangle** widget and drop it at coordinates (650,374). In the toolbar, change **w:** to `262` and **h:** to `70`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ShareBackground`.
5. Repeat step 4 using the following table (* If applicable):

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
Label	(800,140)	`prev` (Note: Keep font to 12 and text color to `#999999`)	`25`	`14`	`PreviousLink`
Label	(834,140)	`View All` (Note: Keep font to 12 and text color to `#999999`) (Note: To add the **OnClick** interaction, under **Click to add actions**, in the **Links** drop-down menu under the **Open Link** drop-down menu, click on **Current Window**. Under **Configure actions**, click on **Gallery**. Click on **OK**.)	`42`	`14`	`ViewAllLink`
Label	(886,140)	`next` (Note: Keep font to 12 and text color to `#999999`)	`24`	`14`	`NextLink`
Image	(170,166)		`440`	`344`	`EntryImage`
Label	(650,199)	`My Awesome Photo` (Note: Keep font to 28)	`248`	`33`	`PhotoTitleLabel1`
Label	(650,232)	`The Best Day Ever!` (Note: Keep font to 28)	`244`	`34`	`PhotoTitleLabel2`

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
Label	(650,269)	`http://t.cx/a5TGfd89yoJl` (Note: Keep font to 11 and text color to `#0066FF`)	`118`	`15`	`EntryLink`
Label	(650,301)	`By:` (Note: Keep font to 16 and text color to `#999999`)	`28`	`28`	`ByLabel`
Label	(678,301)	`FirstName L` (Note: Keep font to 16 and text color to `#999999`)	`88`	`28`	`FirstNameLastInitial`
Label	(658,384)	`Share with friends!` (Note: Keep font to 16 and text color to `#999999`)	`133`	`28`	`ShareCopy`
Image	(660,410)		`24`	`24`	`FacebookIcon`
Image	(688,410)		`24`	`24`	`TwitterIcon`

Note

Facebook and Twitter brand assets can be downloaded from https://www.facebookbrand.com/ and https://about.twitter.com/press/brand-assets respectively.

With the Base view designed, we are now ready to update the design of each adaptive view for the 768 and 320 views.

To update the 768 view, click on the 768 button 768 located in the upper-left corner of the design area. With the **View Entry** page still open in the design area, reposition and resize widgets to complete the 768 view using the following table (* If applicable):

Named widget (In the Widget Interactions and Notes pane)	Base view coordinates	New coordinates*	New width* (w:)	New height* (h:)
ShareBackground	(650,374)	(510,430)	250	
PreviousLink	(800,140)	(650,196)		
ViewAllLink	(834,140)	(684,196)		
NextLink	(886,140)	(736,196)		
EntryImage	(170,166)	(30,222)		
PhotoTitleLabel1	(650,199)	(510,255)		
PhotoTitleLabel2	(650,232)	(510,288)		
EntryLink	(650,269)	(513,324)		
ByLabel	(650,301)	(510,360)		
FirstNameLastInitial	(678,301)	(538,360)		
ShareCopy	(658,384)	(518,440)		
FacebookIcon	(660,410)	(520,466)		
TwitterIcon	(688,410)	(548,466)		

To update the 320 view, click on the 320 button 

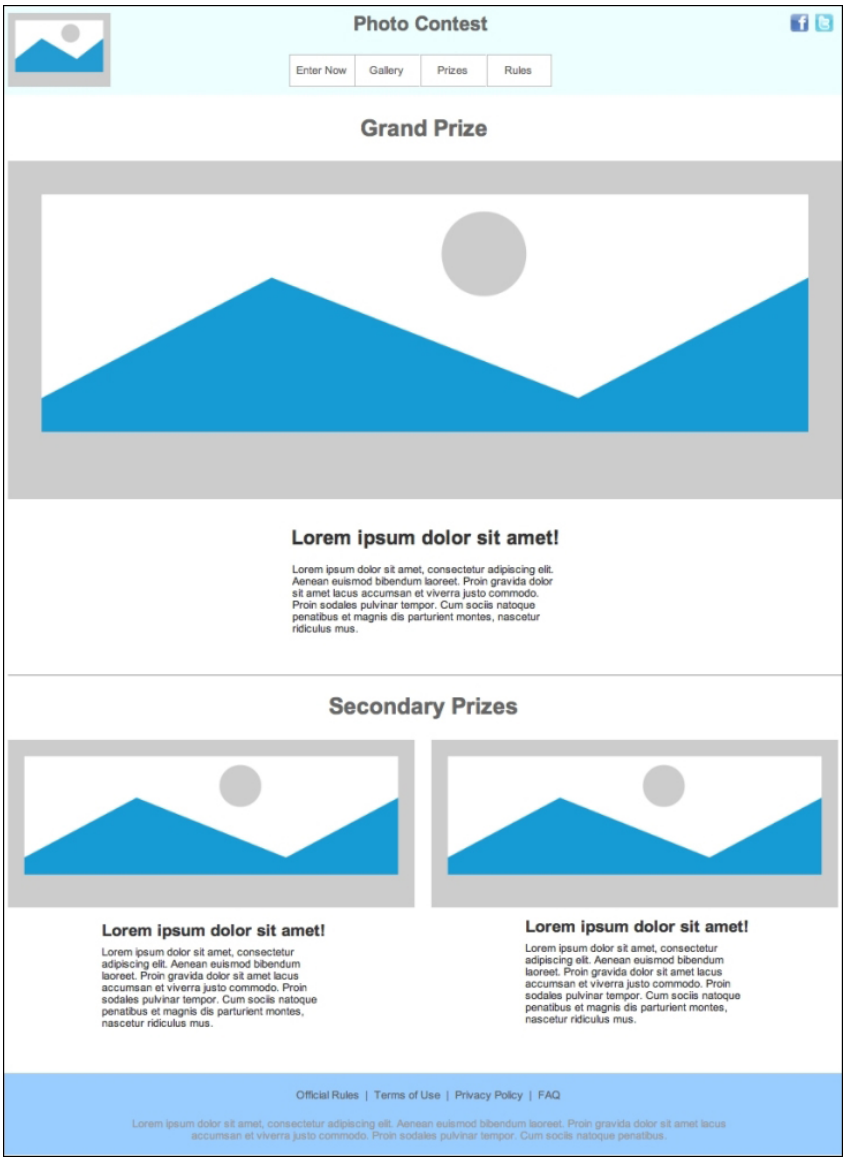
located in the upper-left corner of the design area. With the **View Entry** page still open in the design area, reposition and resize widgets to complete the 320 view using the following table (* If applicable):

Named widget (In the Widget Interactions and Notes pane)	768 View coordinates	New coordinates*	New width* (w:)	New height* (h:)
ShareBackground	(510,430)	(38,630)		
PreviousLink	(650,196)	(210,140)		
ViewAllLink	(684,196)	(244,140)		
NextLink	(736,196)	(296,140)		
EntryImage	(30,222)	(0,166)	320	266
PhotoTitleLabel1	(510,255)	(38,455)		
PhotoTitleLabel2	(510,288)	(38,488)		
EntryLink	(513,324)	(38,526)		
ByLabel	(510,360)	(38,562)		
FirstNameLastInitial	(538,360)	(66,562)		
ShareCopy	(518,440)	(46,640)		
FacebookIcon	(520,466)	(46,665)		
TwitterIcon	(548,466)	(74,665)		

With the design and interactions completed for the **View Entry** page, we are now ready to design the Prizes page.

Designing the Prizes page

Once complete, the **Prizes** page will look as follows:



To design the **Prizes** page for our photo contest, perform the following steps:

1. In the sitemap, double-click on the **Prizes** page to open it in the design area. Select the **Base** Adaptive view.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. From the **Masters** pane, drag the **Footer** master and drop it at coordinates (0,1300). Click on the 768 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `970`. Click on the 320 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `1193`.
4. Click on the **Base** button located in the upper-left corner of the design area. From the **Widgets** pane, drag the **Image widget** and drop it at coordinates (10,190). In the toolbar, change **w:** to `1014` and **h:** to `412`. In the **Widget Interactions and Notes** pane, click in the **Image Name** field and type `GrandPrizeImage`.
5. Repeat step 4 using the following table (* If applicable):

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
H2 Heading 2	(438,135)	Grand Prize			GrandPrizeTitle
H2 Heading 2	(355,635)	Lorem ipsum dolor sit amet!			GrandPrizeHeadline
Paragraph	(356,680)	(Note: Adjust paragraph content to fit.)	321	90	GrandPrizeCopy
Horizontal Line	(10,810)		1014		Line
H2 Heading 2	(400,838)	Secondary Prizes			SecondPrizesTitle
Image	(10,895)		495	204	SecondPrize1Image
H2 Heading 2	(124,1115)	Lorem ipsum dolor sit amet!			SecondPrize1Headline
Paragraph	(124,1145)	(Note: Adjust paragraph content to fit.)	267	115	SecondPrize1Copy
Image	(525,895)		495	204	SecondPrize2Image
H2	(639,1115)	Lorem ipsum dolor sit amet!			SecondPrize2Headline

Widget	Coordinates	Text* (Shown on Widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
Heading 2					
 Paragraph	(639,1145)	(Note: Adjust paragraph content to fit.)	267	115	SecondPrize2Copy

With the Base view designed, we are now ready to update the design of each adaptive view for the 768 and 320 views

To update the 768 view, click on the 768 button 

located in the upper-left corner of the design area. With the **Prizes** page still open in the design area, reposition and resize widgets to complete the 768 view using the following table (* If applicable):

Named widget (In the Widget Interactions and Notes pane)	Base view coordinates	New coordinates*	New width* (w:)	New height* (h:)	New font*
GrandPrizeTitle	(438,135)	(306,128)			
GrandPrizeImage	(10,190)	(10,180)	750	190	
GrandPrizeHeadline	(355,635)	(223,380)			
GrandPrizeCopy	(356,680)	(225,425)	324	83	12
Line	(10,810)	(10,530)	750		
SecondPrizesTitle	(400,838)	(268,560)			
SecondPrize1Image	(10,895)	(10,618)	361	163	
SecondPrize1Headline	(124,1115)	(56,807)			
SecondPrize1Copy	(124,1145)	(58,843)	270	103	12

Named widget (In the Widget Interactions and Notes pane)	Base view coordinates	New coordinates*	New width* (w:)	New height* (h:)	New font*
SecondPrize2Image	(525,895)	(400,618)	361	163	
SecondPrize2Headline	(639,1115)	(446,803)			
SecondPrize2Copy	(639,1145)	(448,843)	270	103	12

To update the 320 view, click on the 320 button 

located in the upper-left corner of the design area. With the **Prizes** page still open in the design area, reposition and resize widgets to complete the 320 view using the following table (* If applicable):

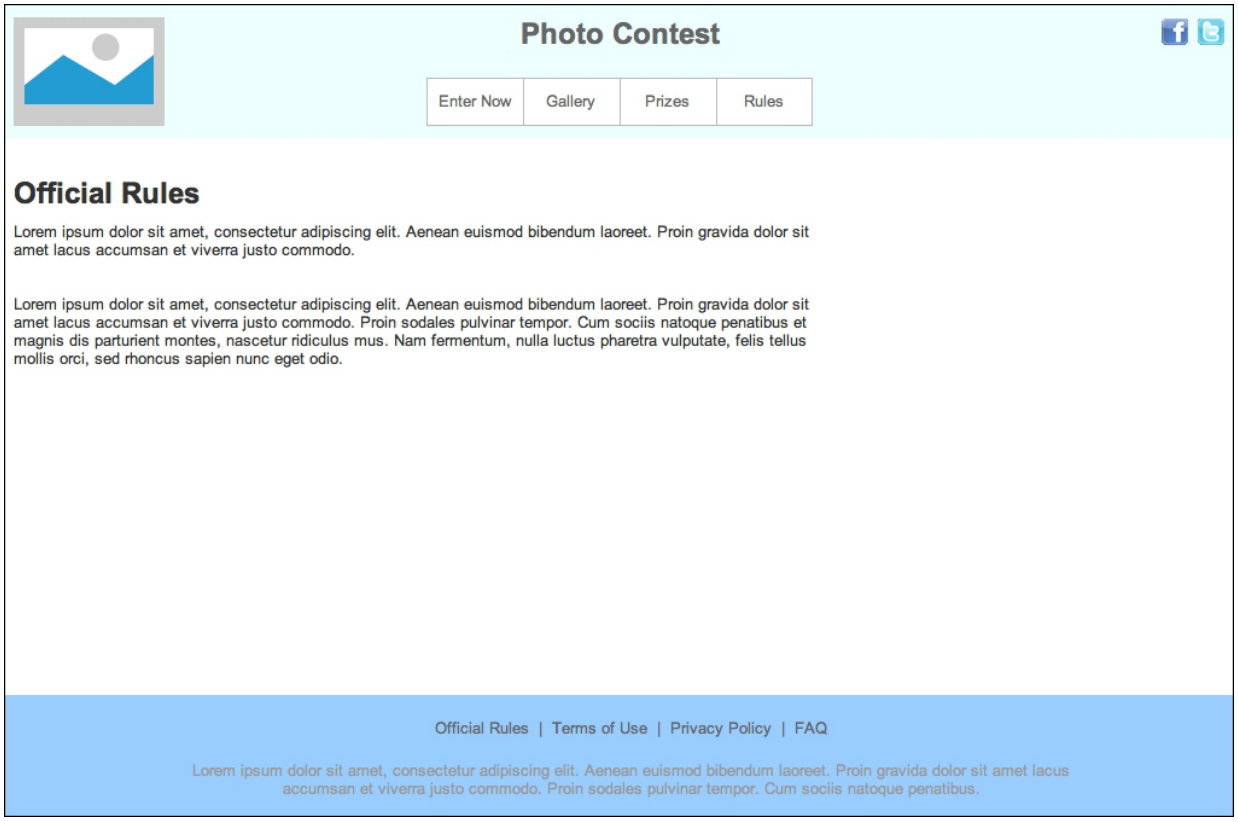
Named widget (In the Widget Interactions and Notes pane)	768 view coordinates	New coordinates*	New width* (w:)	New height* (h:)	New font*
GrandPrizeTitle	(306,128)	(81,140)			
GrandPrizeImage	(10,180)	(5,180)	310	175	
GrandPrizeHeadline	(223,380)	(5,374)	248	23	20
GrandPrizeCopy	(225,425)	(10,414)	315	71	11
Line	(10,530)	(5,495)	310		
SecondPrizesTitle	(268,560)	(43,518)			
SecondPrize1Image	(10,618)	(5,563)	310	175	
SecondPrize1Headline	(56,807)	(5,752)			
SecondPrize1Copy	(58,843)	(5,788)	315	71	11

Named widget (In the Widget Interactions and Notes pane)	768 view coordinates	New coordinates*	New width* (w:)	New height* (h:)	New font*
SecondPrize2Image	(400,618)	(5,882)	310	175	
SecondPrize2Headline	(446,803)	(5,1067)			
SecondPrize2Copy	(448,843)	(5,1102)	315	71	11

With the design and interactions completed for the Prizes page, we are now ready to design the **Rules** page.

Designing the Rules page

Once completed, the **Rules** page will look as follows:



To design the **Rules** page for our photo contest, perform the following steps:

1. In the sitemap, double-click on the **Rules** page to open it in the design area. Select the **Base Adaptive view**.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.

3. From the **Masters** pane, drag the **Footer** master and drop it at coordinates (0,571). Click on the 768 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `875`. Click on the 320 button  located in the upper-left corner of the design area. In the toolbar, change **y:** to `448`.
4. Click on the **Base** button located in the upper-left corner of the design area. From the **Widgets** pane, drag and drop the **Heading 2** widget at coordinates (10,142). In this widget, type `Official Rules`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `RulesHeading`.
5. From the **Widgets** pane, drag and drop the **Paragraph** widget at coordinates (10,180). In the toolbar, change **w:** to `660` and **h:** to `30`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `RulesCopyBlock1`. Adjust the copy shown on the widget to two lines of text.
6. From the **Widgets** pane, drag and drop the **Paragraph widget** at coordinates (10,240). In the toolbar, change **w:** to `660` and **h:** to `60`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `RulesCopyBlock2`.

With the Base view designed, we are now ready to update the design of each adaptive view for the 768 and 320 views.

To update the 768 view, click on the 768 button  located in the upper-left corner of the design area. With the **Rules** page still open in the design area, reposition and resize widgets to complete the 768 view using the following table (* If applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**Base view coordinates**	**New coordinates***	**New width*** **(w:)**	**New height*** **(h:)**	**New font***
RulesCopyBlock1	(10,180)		`540`		
RulesCopyBlock2	(10,240)		`540`	`75`	

To update the 320 view, click on the 320 button  located in the upper-left corner of the design area. With the **Rules** page still open in the design area, reposition and resize widgets to complete the 320 view using the following table (* If applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**768 view coordinates**	**New coordinates***	**New width*** **(w:)**	**New height*** **(h:)**	**New font***
RulesHeading	(10,142)	(5,142)			
RulesCopyBlock1	(10,180)	(5,180)	`294`	`60`	

Named widget (In the Widget Interactions and Notes pane)	768 view coordinates	New coordinates*	New width* (w:)	New height* (h:)	New font*
RulesCopyBlock2	(10,240)	(5,255)	294	135	

Congratulations! We have finished our **Photo Contest** prototype. We can now publish to AxShare and view the **Photo Contest** prototype on our mobile devices through the mobile browser or the AxShare mobile app.

Summary

In this chapter, we created a photo contest optimized for iPad and iPhone devices by using Adaptive Views. We created our Base view with 1,024 width to cover the iPad landscape orientation. We also created a view at 768 width for iPad portrait and a view at 320 width for the iPhone. The photo contest comprised **Home**, registration flow (that is, **Enter**, **Upload Photo**, and **Entry Confirmation**), **Gallery**, and **Entry Detail** pages.

We leveraged a Repeater widget to create the content for the **Gallery** page and used several global variables to store values from a repeater item when a user clicked on a given repeater item. We then displayed the **Entry Detail** page, updating widgets based on the data stored in our global variables.w

In the next chapter, we will explore building a sample e-commerce site, also optimized for tablet and mobile interactions.

Chapter 9. Creating an E-commerce Shopping Cart

Upon reviewing the patterns we have explored so far, we will find that we have the tools needed to create e-commerce home pages, category pages, and product detail pages. Another blueprint to add to our toolbox is an e-commerce shopping cart. We will continue our exploration of Adaptive Views by creating an e-commerce shopping cart tailored for the iPad and iPhone. To accomplish this, we will again leverage two Adaptive Views:

- Portrait tablet (768 x any and below)
- Portrait phone (320 x any and below)

As we did earlier, we will design our Base view targeting a landscape Tablet view of 1,024 pixels (that is, optimized to be backwards compatible with the original iPad). Our e-commerce shopping cart will look as follows:

Cart Subtotal (9 items): 828

Proceed to checkout



Alta 3D Printer

399.00

1 Delete



Arduino Starter Kit

100.00

2 Delete



PLA 3D Printer Filament

33.00

3 Delete



Swimming Pool Leaf Rake

50.00

1 Delete



Aquarium Kit - 5 Gallon

40.00

2 Delete

Search

Home

Cart (#)

My Account

Full Site | Terms of Use | Privacy Policy | FAQ

Copyright 2020 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar tempor. Cum sociis natoque penatibus.

In this chapter, we will cover the following topics:

- Designing an e-commerce shopping cart
- Creating e-commerce shopping cart interactions

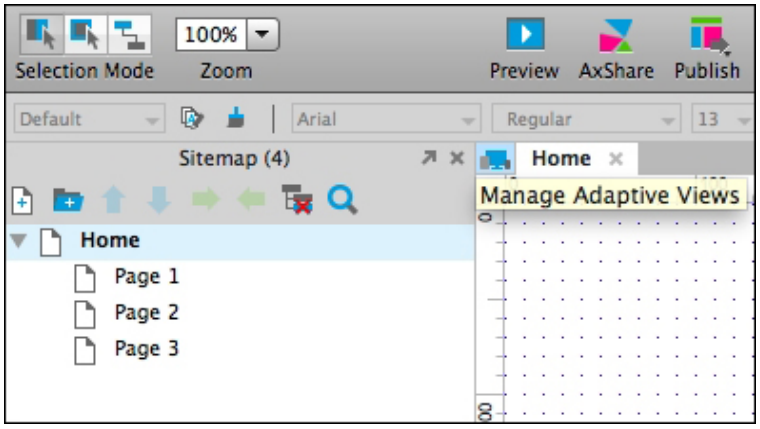
Designing an e-commerce shopping cart

We will start by creating a shopping cart page in our sitemap. We will then configure our Adaptive Views. Next, we will create a global header and footer. The **Shopping Cart** page will comprise of a header, footer, and a cart Repeater widget.

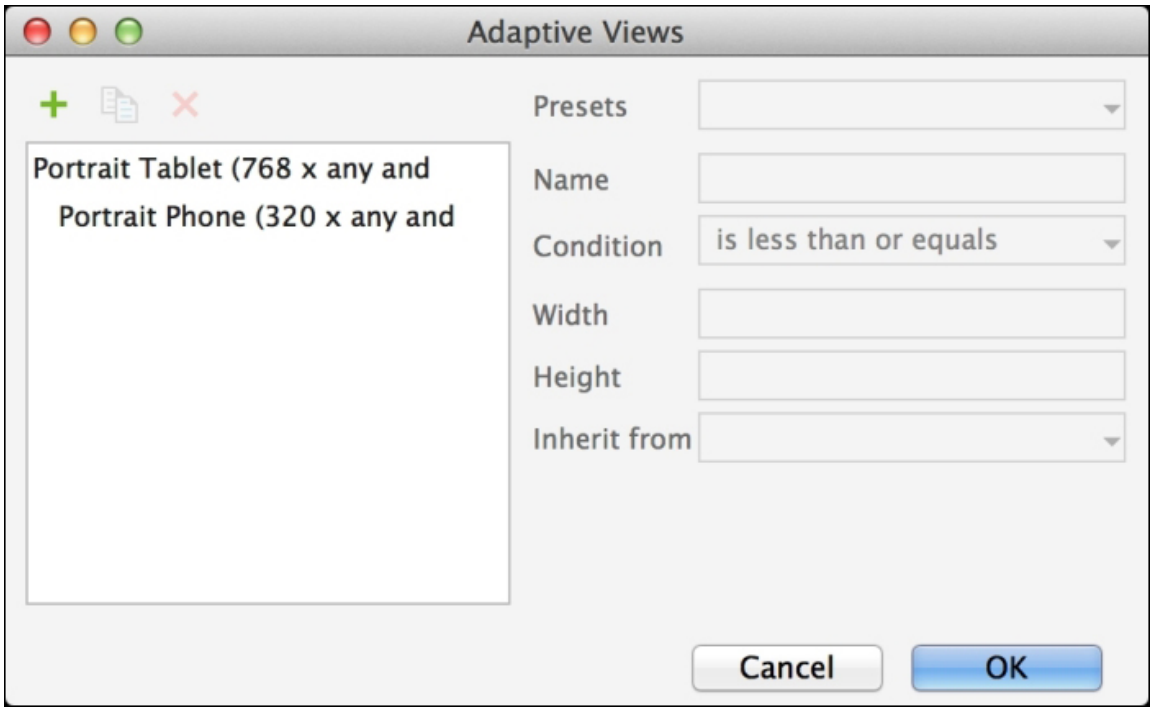
Updating the sitemap and configuring Adaptive Views

To update the sitemap and configure the Adaptive Views for our photo contest, perform the following steps:

1. Open a new Axure document and organize the sitemap with a single page, **Shopping Cart**
2. To configure Adaptive Views, click on the **Manage Adaptive Views** icon located in the upper-left corner of the design area as shown in the following screenshot:



3. In the Adaptive Views dialog box, click on the green **+** sign, and from the **Presets** drop-down menu, select **Portrait Tablet**. Click again on the green **+** sign, and from the **Presets** drop-down menu, select **Portrait Phone**. Click on **OK**. Once completed, your **Adaptive Views** dialog box should look like this:



With the sitemap updated and the adaptive views configured, we will now create global variables as well as the design for our **Header** and **Footer** masters.

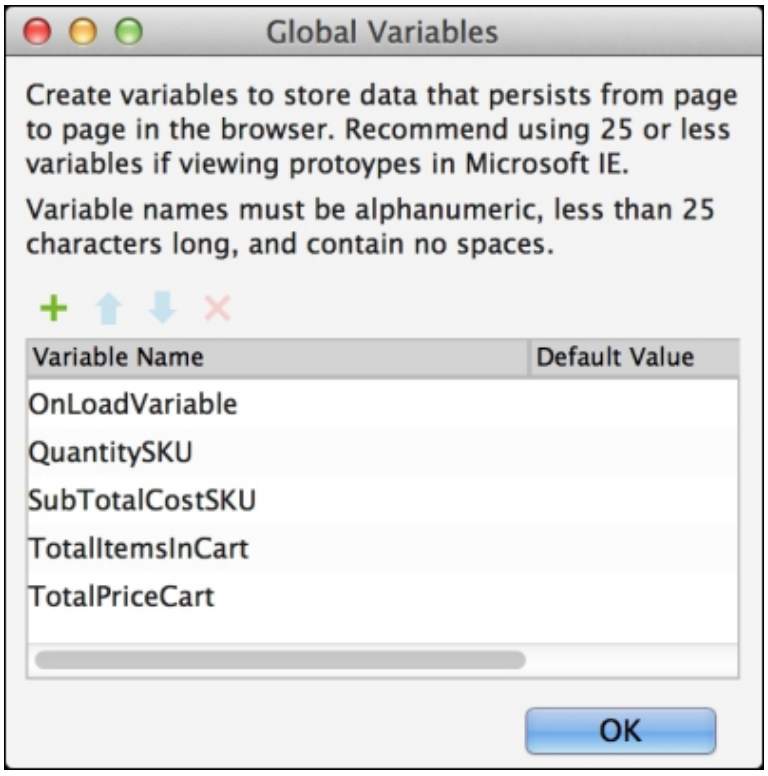
Creating global variables

We will now create global variables, which will allow us to share data between pages in our design. To create global variables, in the main menu, click on **Project** and then on **Global Variables…**. In the **Global Variables** dialog, perform the following steps:

1. Click on the green **+** sign and type `QuantitySKU`. Leave the **Default Value** field empty.
2. Repeat step 1 three more times to create additional variables using the following table for the **Variable Name** field:

Variable Name
SubTotalCostSKU
TotalItemsInCart
TotalPriceCart

Once completed, the **Global Variables** dialog will look like this:



3. Click on **OK**.

With our global variables created, we are now ready to complete the design and interactions for the **Header** and **Footer** masters. We will start by adding masters to the **Masters** pane.

Adding masters to the Masters pane

To create masters, perform the following steps:

1. In the **Masters** pane, click on the Add Master icon, type `Header`, and press *Enter*.
2. Next, right-click on the icon next to the **Header** master, mouse over **Drop Behavior**, and click on **Lock to Master Location**.
3. Finally, click on the Add Master icon, type `Footer`, and press *Enter*.

With our masters added to the **Masters** pane, we are now ready to complete the design and interactions for each individual master. We will start with the **Header** master.

Designing the Header master

Once completed, the **Header** master will look as follows:



We will now create a solid color background for the header that will stretch to cover the width of the browser window. We will use a dynamic panel leveraging a back color with **Background Repeat** set for **Stretch to Cover**. To do this, perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Header** master to open in the design area.
2. From the **Widgets** pane, drag the **Dynamic Panel** widget and place it at coordinates (0,0). In the toolbar, change **w:** to `1024` and **h:** to `50`. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `HeaderBackground`.
3. With the **HeaderBackground** dynamic panel still selected, click on the 768 button 

located in the upper-left corner of the design area. In the toolbar, change **w:** to `768`.

4. With the **HeaderBackground** dynamic panel still selected, click on the 320 button  located in the upper-left corner of the design area. In the toolbar, change **w:** to `320`.
5. Click on the **Base** button located in the upper-left corner of the design area. In the **Widget Manager** pane, double-click on **State1** to open in the design area. Under the design area with the **Panel State Style** tab selected, select a **Back Color** and set **Repeat to Stretch to Cover**.

To place widgets for the **Header** master with **State1** open in the design area, select the **Base** Adaptive View and perform the following steps:

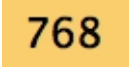
1. From the **Widgets** pane, drag the **Image** widget  and place it at coordinates (10,8). In the toolbar, change **w:** to `100` and **h:** to `35`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Logo`.
2. Repeat step 2 to complete the **Base** view of the **Header** master using the following table (* If applicable):

Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
 Image	(930,10)		`30`	`30`	`SearchIcon`
 Image	(975,10)	(Note: To add the **OnClick** interaction, under **Click to add**	`40`	`30`	`ShoppingCartIcon`

Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
		actions, in the **Links** drop-down menu under the **Open Link** drop-down menu, click on **Current Window**. Under **Configure actions**, click on **Shopping Cart**. Click on **OK**.)			

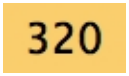
Note

For the search (magnifying glass) and shopping cart icons, you could utilize images, emojis (http://emojipedia.org), or font awesome icons (http://fortawesome.github.io/Font-Awesome/cheatsheet/). To enhance your prototypes with Font Awesome, check out http://www.axure.com/c/forum/tips-tricks-examples/8732-font-awesome-widget-library-v7-icon-fonts.html.

To update widgets for the **Header** master in the portrait tablet view with **State1** open in the design Area, click on the 768 button  located

in the upper-left corner of the design area. With the **Header** master open in the design area, reposition and resize widgets to complete the 768 view using the following table (* if applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**Base View coordinates**	**New coordinates***
SearchIcon	(930,10)	(680,10)
ShoppingCartIcon	(975,10)	(720,10)

To update widgets for the **Header** master in the portrait mobile view with **State1** open in the design area, click on the 320 button 

located in the upper-left corner of the design area. With the **Header** master open in the design area, reposition and resize widgets to complete the 320 view using the following table (* if applicable):

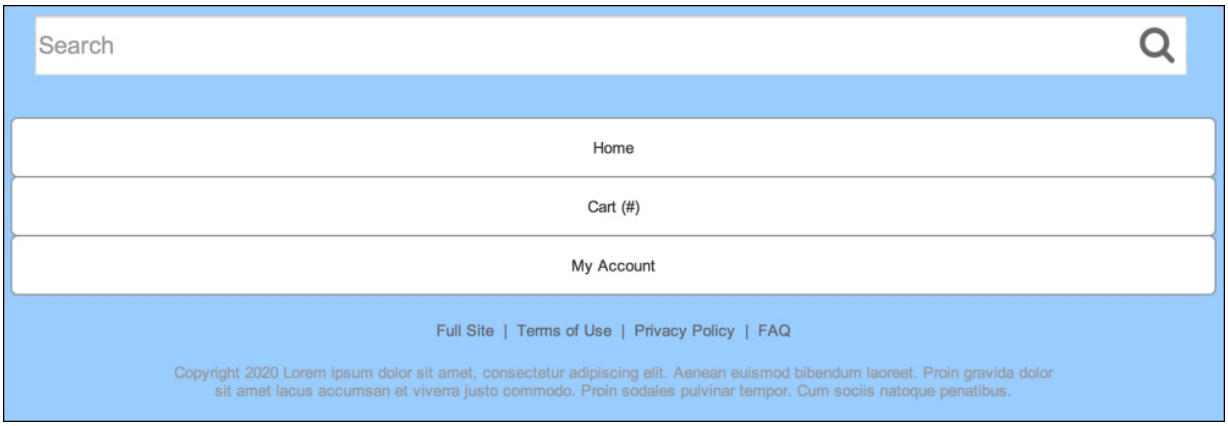
Named widget **(In the Widget Interactions and Notes pane)**	**768 view coordinates**	**New coordinates***
Logo	(10,8)	(5,8)
SearchIcon	(680,10)	(230,10)

Named widget (In the Widget Interactions and Notes pane)	768 view coordinates	New coordinates*
ShoppingCartIcon	(720,10)	(270,10)

We will now design our **Footer** master.

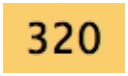
Designing the Footer master

Once completed, the **Footer** master will look as follows:



We will now create a solid color background for the footer that will stretch to cover the width of the browser window. We will use a dynamic panel leveraging a back color with **Background Repeat** set for **Stretch to Cover**. To do so, perform the following steps:

1. In the **Masters** pane, double-click on the icon next to the **Footer** master to open in the design area.
2. From the **Widgets** pane, drag the **Dynamic Panel** widget and place it at coordinates (0,1150). In the toolbar, change **w:** to `1024` and **h:** to `350`. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `FooterDP`.

3. With the **FooterDP** dynamic panel still selected, click on the 768 button  located in the upper-left corner of the design area. In the toolbar, change **x:** to `0`, **y:** to `918`, **w:** to `768`, and **h:** to `352`.
4. With the **FooterDP** dynamic panel still selected, click on the 320 button  located in the upper-left corner of the design area. In the toolbar, change **w:** to `320` and **h:** to `372`.
5. Click on the **Base** button located in the upper-left corner of the design area. In the **Widget Manager** pane, double click on **State1** to open in the design area. Under the design area with the **Panel State Style** tab selected, select a **Back Color** and set **Repeat to Stretch to Cover**.

To place widgets for the **Footer** master with **State1** open in the design area, select the **Base** Adaptive View and perform the following steps:

1. From the **Widgets** pane, drag the **Label** widget  and place it at coordinates (364,265). Type `Full Site | Terms of Use | Privacy Policy | FAQ` in the widget. Next, in the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `FooterLinks`.
2. From the **Widgets** pane, drag the **Label** widget and place it at coordinates (142,300). Type Greek placeholder text to be displayed on the Label widget. In the toolbar, change **w:** to `740` and **h:** to `30`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `FooterLegal`. In the **Widget Properties and Style** pane, with the **Style** tab selected, scroll to **Font** and perform the following steps:

    1. Click on the down arrow on the text color icon . In the drop-down menu, in the **#** text field, enter `999999`.
    2. Under the **Alignment + Padding** dropdown, click on the Align Center icon .
    3. Type `Copyright 2020`. Type additional placeholder legal copy.

3. Repeat step 2 using the following table (* If applicable):

Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
 Text Field	(30,10)	(Note: To add placeholder text, in the **Widget Properties and Style** pane, click on the **Properties** tab. In the **Hint Text** field, type `Search`. Click on **Hint Style** and update the font size to 20 and text color to `#999999`. Click on **OK**.)	`960`	`50`	`FooterSearchField`
 Image	(950,20)	(Note: Update the image with a Search icon.)	`30`	`30`	`FooterSearchIcon`

Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
 Button Shape	(10,95)	Home	1004	50	HomeMenuItem
 Button Shape	(10,144)	Cart (#) (Note: To add the **OnClick** interaction, under **Click to add actions**, in the **Links** drop-down menu under the **Open Link** drop-down menu, click on **Current Window**. Under **Configure actions**, click on **Shopping**	1004	50	CartMenuItem

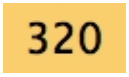
Widget	Coordinates	Text* (Shown on widget)	Width (w:)	Height (h:)	Name field (In the Widget Interactions and Notes pane)
		Cart. Click on **OK**.)			
 Button Shape	(10,193)	My Account	1004	50	MyAccountMenuItem

To update widgets for the Footer master in the portrait tablet view with **State1** open in the design area, click on the 768 button 

located in the upper-left corner of the design area and reposition and resize widgets to complete the 768 view using the following table (* if applicable):

Named widget (In the Widget Interactions and Notes pane)	Base View coordinates	New coordinates*	New width* (w:)	New height* (h:)
FooterSearchField	(30,10)		710	
FooterSearchIcon	(950,20)	(700,20)		
HomeMenuItem	(10,95)		750	

Named widget (In the Widget Interactions and Notes pane)	Base View coordinates	New coordinates*	New width* (w:)	New height* (h:)
CartMenuItem	(10,144)		750	
MyAccountMenuItem	(10,193)		750	
FooterLinks	(364,265)	(221,268)		
FooterLegal	(142,300)	(102,298)	564	42

To update widgets for the **Footer** master in the portrait mobile view with **State1** open in the design area, click on the 320 button 

located in the upper-left corner of the design area, and reposition and resize widgets to complete the 320 view using the following table (* if applicable):

Named widget (In the Widget Interactions and Notes pane)	768 View coordinates	New coordinates*	New width* (w:)	New height* (h:)
FooterSearchField	(30,10)	(20,10)	280	

Named widget (In the Widget Interactions and Notes pane)	768 View coordinates	New coordinates*	New width* (w:)	New height* (h:)
FooterSearchIcon	(700,20)	(260,20)		
HomeMenuItem	(10,95)	(10,76)	300	
CartMenuItem	(10,144)	(10,125)	300	
MyAccountMenuItem	(10,193)	(10,174)	300	
FooterLinks	(221,268)	(29,246)		
FooterLegal	(102,298)	(42,276)	237	84

With the **Header** and **Footer** masters complete, we are now ready to design the **Shopping Cart** page.

Designing the Shopping Cart page

Once completed, the **Shopping Cart** page will look as follows:

Cart Subtotal (9 items): 828

Proceed to checkout



Alta 3D Printer

399.00

1 Delete



Arduino Starter Kit

100.00

2 Delete



PLA 3D Printer Filament

33.00

3 Delete



Swimming Pool Leaf Rake

50.00

1 Delete



Aquarium Kit - 5 Gallon

40.00

2 Delete

Search

Home

Cart (#)

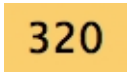
My Account

Full Site | Terms of Use | Privacy Policy | FAQ

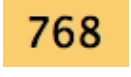
Copyright 2020 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar tempor. Cum sociis natoque penatibus.

To design the **Shopping Cart** page, perform the following steps:

1. I n the sitemap, double-click on the **Shopping Cart** page to open it in the design area. Select the **Base** Adaptive View.
2. From the **Masters** pane, drag the **Header** master and drop it anywhere in the design area.
3. Next, drag the **Footer** master and drop it at coordinates (0,1030). In the **Widget Interactions and Notes** pane, click on the **Footer Name** field and type `CartFooter`. Click on the 768 button 

located in the upper-left corner of the design area. In the toolbar, make sure **y:** to `1030`. Click on the 320 button 

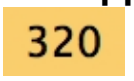
located in the upper-left corner of the design area. In the toolbar, make sure **y:** to `1030`.
4. From the **Widgets** pane, drag the **Label** widget  and place it at coordinates (10,70). Type `Cart Subtotal (# items):` in the widget. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `SubTotalLabel`. In the **Widget Properties and Style** pane with the **Style** tab selected, update the font size to `16`.
5. From the **Widgets** pane, drag the **Label** widget  and place it at coordinates (200,70). Type `000.00` in the widget. Next, in the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `SubTotal`. In the**Widget Properties and Style** pane with the **Style** tab selected, update the font size to `16` and text color  to `#999999`.
6. From the **Widgets** pane, drag the **Label** widget  and place it at coordinates (10,110). Type `Proceed to checkout` in the widget. In the toolbar, change **w:** to `1004` and **h:** to `60`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `CheckoutButton`. In the **Widget Properties and Style** pane with the **Style** tab selected, update the font size to `16`, and under **Borders, Lines + Fills**, update the Fill color  to `#797979`.

To update the label and button shape widgets in the portrait tablet view with the **Shopping Cart** page open in the design area, click on the 768 button 

located in the upper-left corner of the design area, and

reposition and resize widgets to complete the 768 view using the following table (* if applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**Base View coordinates**	**New coordinates***	**New width*** **(w:)**	**New height*** **(h:)**
CheckoutButton	(10,110)		748	

To update the label and button shape widgets in the portrait mobile view with the **Shopping Cart** page open in the design area, click on the 320 button 

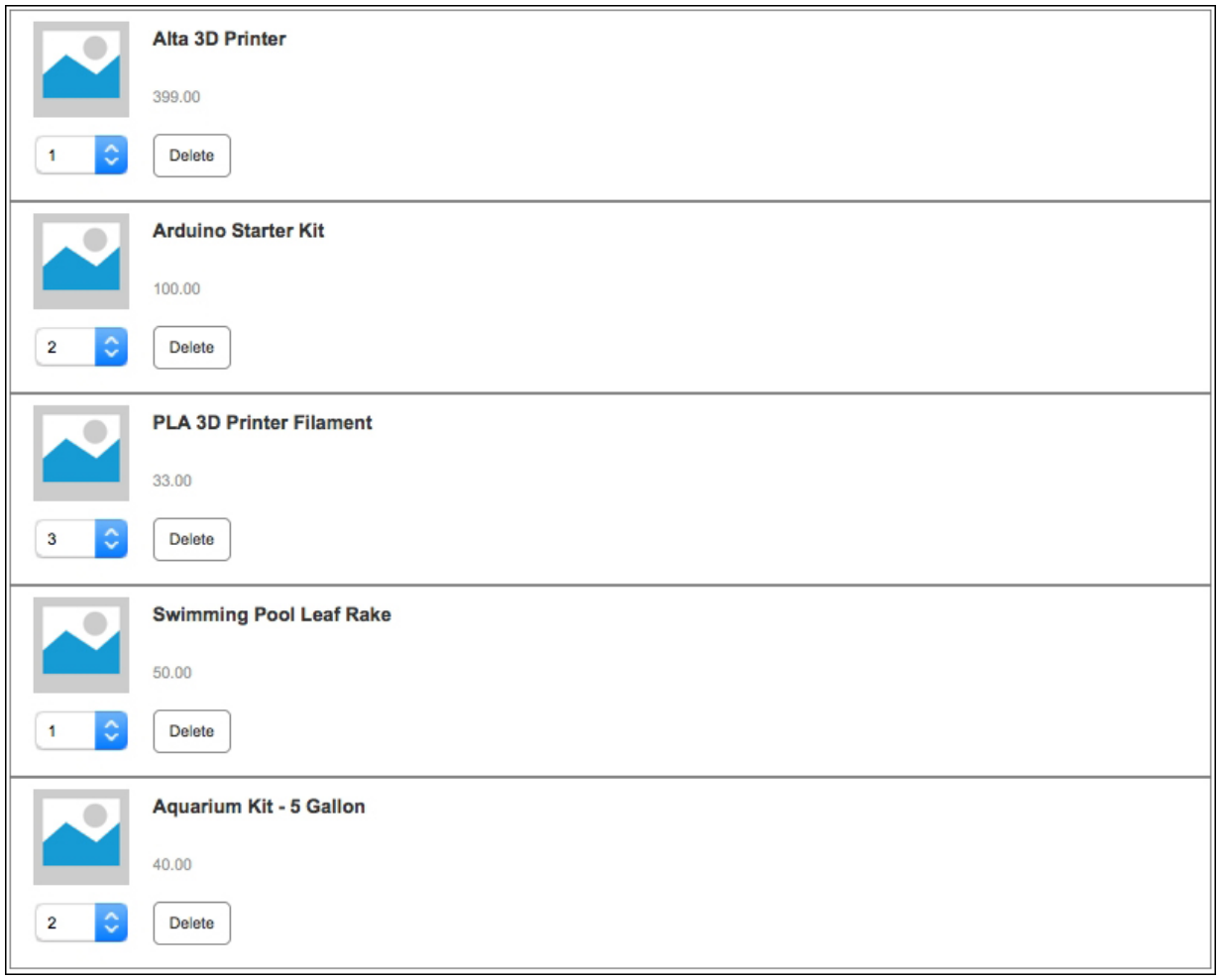
located in the upper-left corner of the design area, and reposition and resize widgets to complete the 320 view using the following table (* if applicable):

Named widget **(In the Widget Interactions and Notes pane)**	**768 View coordinates**	**New coordinates***	**New width*** **(w:)**	**New height*** **(h:)**
CheckoutButton	(10,110)		300	

With the **Header**, **Footer**, and widgets placed, we will now design the Cart repeater.

Designing the Cart repeater

Once completed, our Cart repeater displaying our default dataset will look as follows:

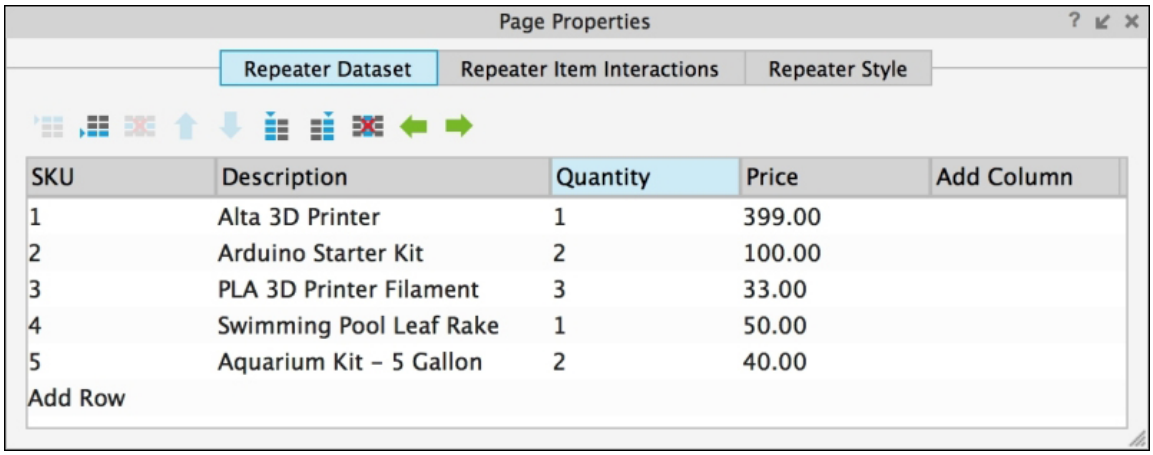


With the Shopping Cart page open in the design area and the Base view selected, perform the following steps:

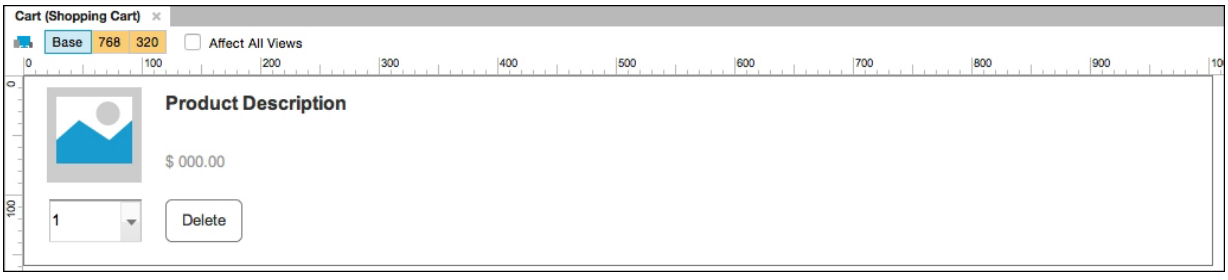
1. From the **Widgets** pane, drag the **Repeater** widget into the design area and place it at coordinates (10,200). In the **Widget Interactions and Notes** pane, click on the **Repeater Footnote and Name** field and type `Cart`.
2. Double-click on the Cart repeater widget to open it in the design area.
3. In the Repeater pane under the design area, click on the **Repeater Style** tab. To change the default **Repeater Style**, under the **Layout** dropdown, click on the radio button next to **Vertical**.

4. Under the design area in the Repeater pane, click on the **Repeater Dataset** tab. Use the values given in the following table to update data values:

> ## Note
>
> To rename columns or rows, double-click the column or row and type the new column or row name. To add new columns or rows, click on **add column** or **add row**.



Once completed, the **Cart** repeater item will look as follows:



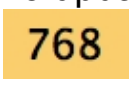
We will now resize the Rectangle widget on the Repeater item and place additional widgets. To do so, perform the following steps:

1. With the repeater still open in the design area, click on the **Rectangle** widget at coordinates (0,0). In the toolbar, change **w:** to `1004` and **h:** to `160`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ProductBackground`.

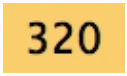
2. From the **Widgets** pane, drag the **Image** widget  and place it at coordinates (20,10). In the toolbar, change **w:** to `80` and **h:** to `80`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ProductImage`.
3. From the **Widgets** pane, drag the **Label** widget  and place it at coordinates (120,15). Type `Product Description` in the widget. In the toolbar, change**w:** to `470` and **h:** to `40`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `ProductDescription`. In the **Widget Properties and Style** pane with the **Style** tab selected, update the font size to `16`.
4. From the **Widgets** pane, drag the **Label** widget  and place it at coordinates (120,65). Type `$ 000.00` in the widget. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Price`. In the toolbar, change **w:** to `51` and **h:** to `15`. With the **Style** tab selected, update the text color  to `#999999`.
5. From the **Widgets** pane, drag the **Droplist** widget  and place it at coordinates (22,104). In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `Quantity`. Right-click on the **Droplist** widget and click on **Edit List Items**. In the **Edit Droplist** dialog, click on the **Add Many** button and add the numbers 1-10. Click on **OK**.
6. From the **Widgets** pane, drag the **Button Shape** widget  and place it at coordinates (120,104). In the toolbar, change **w:** to `65` and **h:** to `36`. Type `Delete`. In the **Widget Interactions and Notes** pane, click on the **Shape Name** field and type `DeleteButton`.

With the Base view designed for the Cart repeater, we are now ready to update the design of each adaptive view for the 768 and 320 views.

Updating the Cart repeater for the 768 and 320 views

To update the cart repeater for the 768 view, click on the 768 button  located in the upper-left corner of the design area. With the Cart repeater item still open in the design area, reposition and resize widgets to complete the 768 view using the following table (* if applicable):

Named widget (In the Widget Interactions and Notes pane)	Base View coordinates	New coordinates*	Width* (w:)	Height* (h:)
ProductBackground	(0,0)		748	

To update the 320 view, click on the 320 button  located in the upper-left corner of the design area. With the Cart repeater item page still open in the design area, reposition and resize widgets to complete the 320 view using the following table (* if applicable):

Named widget (In the Widget Interactions and Notes pane)	768 View coordinates	New coordinates*	Width* (w:)	Height* (h:)
ProductBackground	(0,0)		300	
ProductDescription	(120,15)		170	40

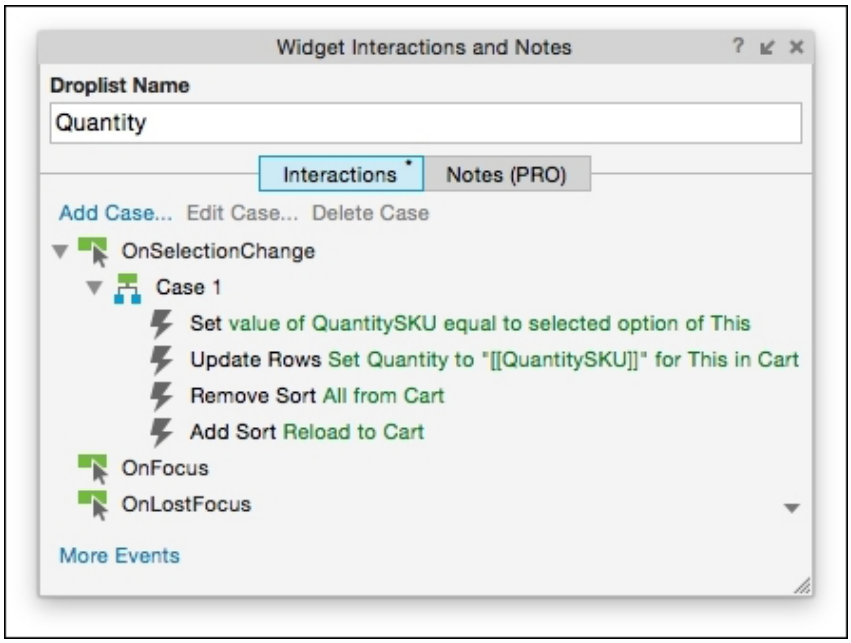
With the layout updated for the Cart repeater, we are now ready to define the interactions.

Defining the Cart repeater interactions

Since we have already defined all of our global variables, we can now complete the interactions for our repeater item. We will define the **OnSelectionChange** event for the **Quantity** droplist and the **OnClick** event for the **DeleteButton** widget.

Creating the Quantity widget's OnSelectionChange case

Once completed, **Case 1** for the **OnSelectionChange** will be defined as follows:



With the repeater item still open in the design area and the **Base** Adaptive View selected, in the **Widget Interactions and Notes** pane, click on the **Quantity** droplist at coordinates (22,104). In the **Interactions** tab, double-click on the **OnSelectionChange** event. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

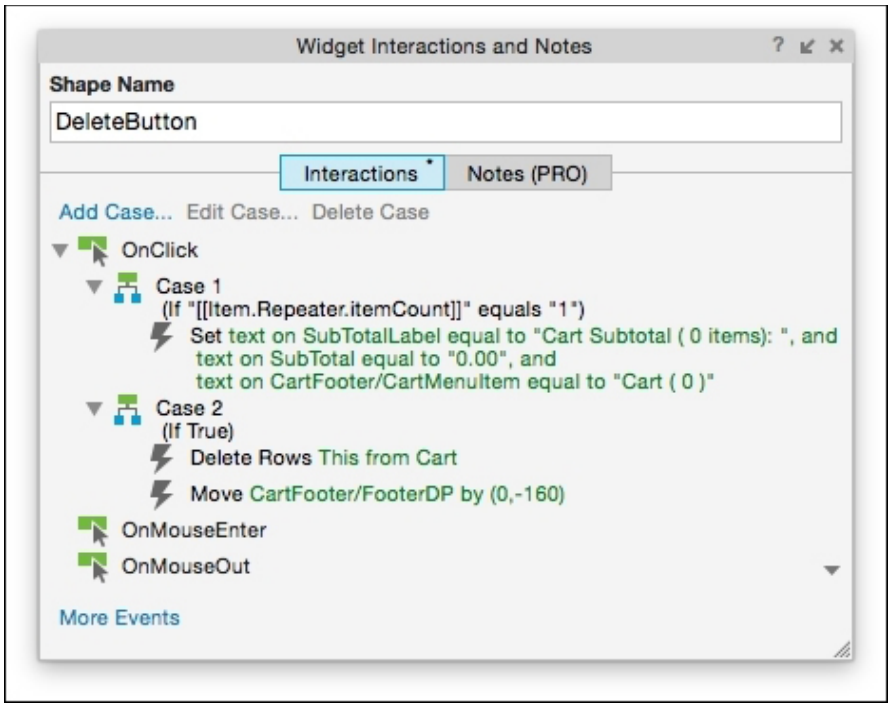
1. Create the first action. To set variable value for key global variables:
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Variable Value**.
    2. Under **Configure actions**, click on the checkbox next to **QuantitySKU**.

    3. Under **Set variable to**, click on the first dropdown and then on **selected option of**. In the text field dropdown, select `This`.
2. Create the second action. To update rows, under **Click to add actions**, in the **Repeaters** drop-down menu under the **Datasets** drop-down menu, click on **Update Rows**. Under **Configure actions**, click on the checkbox next to **Cart**. Click on the **Select Column** dropdown and then on **Quantity**. Click on the **fx** button and in the **Insert Variable or Function…** field, enter `[[QuantitySKU]]`. Click on **OK**.
3. Create the third action. To remove sort, under **Click to add actions**, in the **Repeaters** drop-down menu, click on **Remove Sort**. Under **Configure actions**, click on the checkbox next to **Cart**. Click on the checkbox next to **Remove all sorting**.
4. Create the fourth action. To add sort, under **Click to add actions**, in the **Repeaters** drop-down menu under the **Datasets** drop-down menu, click on **Add Sort**. Under **Configure actions**, click on the checkbox next to **Cart**. Click on the **Name** field and enter `Reload`. The **Property** dropdown equals `SKU`. The **Sort as** drop-down equals `Number`. The **Order** dropdown `Ascending`. Click on **OK**.

With the case defined for the **OnSelectionChange** event, we are now ready to define the case for the **DeleteButton OnClick** event.

Creating the DeleteButton widget's OnClick cases

Once completed, the **Widget Interactions and Notes** pane for the **DeleteButton** will look as follows:



To enable interactions for the **DeleteButton** widget, do the following:

1. With the Cart repeater open in the design area, click on the **DeleteButton** widget at coordinates (120,104).
2. Double-click on the **OnClick** event. A **Case Editor** dialog box will open.
3. Add the condition: Click on the **Add Condition** button.
4. In the outlined condition box within the **Condition Builder** dialog box, perform the following steps:

    1. In the first dropdown, select **value**.
    2. In the second field, enter `[[Item.Repeater.itemCount]]`.
    3. In the third dropdown, select **equals**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth field, enter `1`.

5. Create the actions: To set text, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **SubTotalLabel**. In the **Set text to** dropdown, select **value**, and enter `Cart Subtotal (0 items):` in the field.

3. Under **Configure actions**, click on the checkbox next to **SubTotal**. In the **Set text to** dropdown, select **value** and enter `0.00` in the field.
4. Under **Configure actions**, click on the checkbox next to **CartMenuItem**. In the **Set text to** dropdown, select **value** and enter `Cart (0)` in the field. Click on **OK**.

6. Create **Case 2**. Double-click on the **OnClick** event.
7. Create the first action, to delete rows, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Repeaters** drop-down menu under the **Datasets** dropdown, and click on **Delete Rows**.
    2. Under **Configure actions**, in the **Select the repeaters to delete their items from** section, click on the checkbox next to **Cart**. Click on the radio button next to **This**.

8. Create the second action, to move **CartFooter**/**FooterDP**, perform the following steps:

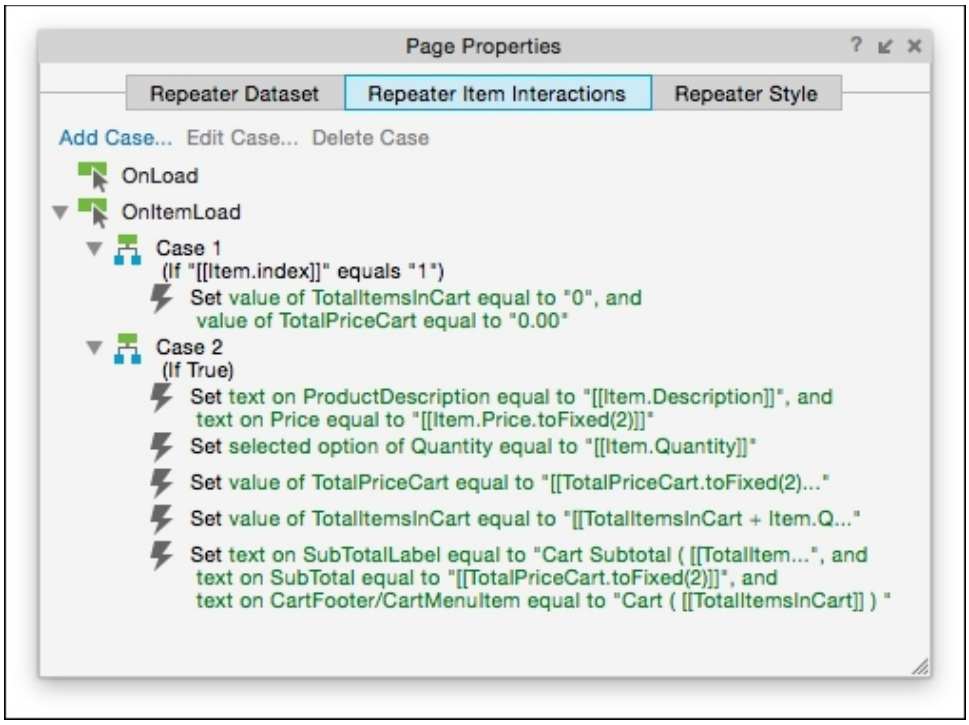
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Move**.
    2. Under **Configure actions**, in the **Select the widgets to move** section, under **CartFooter**, click on the checkbox next to **FooterDP**.
    3. In the **Move:** dropdown, select `by` and enter **y:** to `-160` .

9. Click on **OK**.
10. Right-click on **Case 2** and click **Toggle IF/ELSE IF**.

With the **DeleteButton** widget's **OnClick** cases defined, we are now ready to complete the **OnItemLoad** event for the Cart repeater.

Creating cases for the Cart repeater OnItemLoad event

Once defined, the Cart repeater's **OnItemLoad** event will look as follows:



To complete interactions for the repeater, we also need to update the **OnItemLoad** event. To build the first case, under the design area in the Repeater pane, click on the **Repeater Item Interactions** tab. Double-click on the **OnItemLoad** event. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

1. Add the condition, click on the **Add Condition** button.
2. In the outlined condition box within the **Condition Builder** dialog box, perform the following steps:

    1. In the first dropdown, select **value**.
    2. In the second, enter `[[Item.index]]`.
    3. In the third dropdown, select **equals**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth, enter `1`.
    6. Click on **OK**.

3. Create the first action, to set value for widgets on the repeater item:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu and click on **Set Variable Value**.

2. Under **Configure actions**, in the **Select the variables** to set section, click on the checkbox next to **TotalItemsInCart**. Under **Set variable to** in the dropdown, select **value** and enter `0` in the text field.
3. Under **Configure actions**, in the **Select the variables** to set section, click on the checkbox next to **TotalPriceCart**. Under **Set variable to select** in the dropdown, select **value** and enter `0.00` in the text field.
4. Click on **OK**.

To build the second case, under the design area in the Repeater pane, with the **Repeater Item Interactions** tab selected, double-click on the **OnItemLoad** event. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

1. Create the first action, to set text, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** dropdown menu. Click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **ProductDescription**. In the **Set text to** dropdown, select **value** and enter `[[Item.Description]]` in the field.
    3. Under **Configure actions**, click on the checkbox next to **Price**. In the **Set text to** dropdown, select value and enter `[[Item.Price.toFixed(2)]]` in the field.

2. Create the second action to set selected option:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on **Set Selected List Option**.
    2. Under **Configure actions**, click on the checkbox next to **Quantity**.
    3. Under **Set selected option to** in the dropdown, select **value** and enter `[[Item.Quantity]]` in the text field.

3. Create the third action, to set value, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Variables** drop-down menu. Click on **Set Variable Value**.

2. Under **Configure actions**, click on the checkbox next to **TotalPriceCart**. In the **Set text to** dropdown, select **value** and enter `[[TotalPriceCart.toFixed(2) + (Item.Quantity*Item.Price.toFixed(2))]]]]` in the field.
3. Under **Configure actions**, click on the checkbox next to **TotalItemsInCart**. In the **Set text to** dropdown, select **value** and enter `[[TotalItemsInCart + Item.Quantity]]` in the field.

4. Create the fourth action, to set text, perform the following steps:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu. Click on **Set Text**.
    2. Under **Configure actions**, click on the checkbox next to **SubTotalLabel**. In the **Set text to** dropdown, select **value** and enter `Cart Subtotal ([[TotalItemsInCart]] items):` in the field.
    3. Under **Configure actions**, click on the checkbox next to **SubTotal**. In the **Set text to** dropdown, select value and enter `[[TotalPriceCart.toFixed(2)]]` in the field.
    4. Under **Configure actions**, click on the checkbox next to **CartMenuItem**. In the **Set text to** dropdown, select **value** and enter `Cart ( [[TotalItemsInCart]] )` in the field.

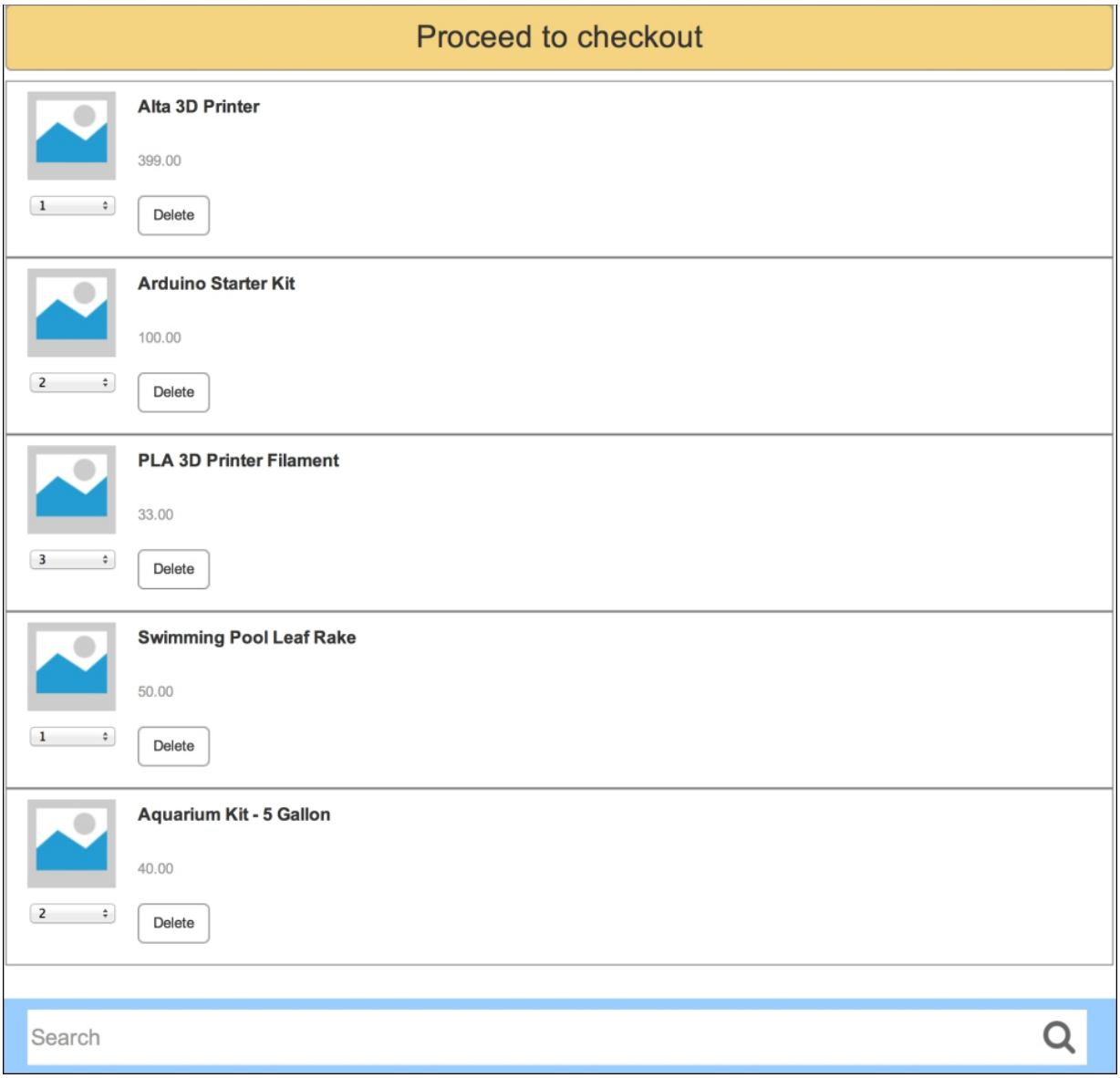
5. Click on **OK**.
6. Right-click on **Case 2** and click on **Toggle IF/ELSE IF**.

With the interactions completed for the repeater, we are now ready to add a sticky **Proceed to Checkout** button.

Creating the sticky Proceed to checkout button

To increase conversion rates, e-commerce sites will often use sticky checkout buttons on the shopping cart. A sticky checkout button will "stick" to the top of the browser when the user scrolls vertically past a given point on the page.

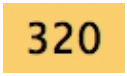
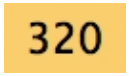
For our **Shopping Cart** page, when the window is scrolled beyond 110 pixels in the *y* direction, a **StickyDP** dynamic panel will be shown, which will stick to the top of the browser. **State1** of **StickyDP** only contains a button widget. The following screenshot shows the Base view of the **Shopping Cart** page scrolled beyond 100 pixels in the *y* direction:



To place the **StickyDP** and **StickyCheckOutbutton**, do the following:

1. In the sitemap, double-click on the **Shopping Cart** page to open it in the design area. Select the **Base** Adaptive View.
2. From the **Widgets**pane, drag the **Dynamic Panel** widget and place it at coordinates (10,0). In the toolbar, change **w:** to `1004` and **h:** to `60`. Click on the checkbox next to **Hidden**. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `StickyDP`.
3. Right-click on the **StickyDP** dynamic panel and click on **Pin to Browser**. In the **Pin to Browser** dialog box, click on the checkbox

next to **Pin to browser window**. Under **Horizontal Pin**, click on the **Center** radio button. Under **Vertical Pin**, click on the **Top** radio button. Click on the checkbox next to **Keep in front (browser only)**. Click on **OK**.

4. With the **StickyDP** dynamic panel still selected, click on the 768 button  located in the upper-left corner of the design area. In the toolbar, change **x:** to `0` and **w:** to `768`.
5. With the **StickyDP** dynamic panel still selected, click on the 320 button  located in the upper-left corner of the design area. In the toolbar, change **w:** to `320`.
6. Click on the **Base** button located in the upper-left corner of the design area. In the **Widget Manager** pane, double-click on **State1** to open in the design area.
7. From the **Widgets** pane, drag and drop the **Button Shape** widget  at coordinates (0,0). In the toolbar, change **w:** to `1004` and **h:** to `60`. In the **Widget Interactions and Notes** pane, click on the **Dynamic Panel Name** field and type `StickyCheckoutButton`. In the **Widget Properties and Style** pane, with the **Style** tab selected, update the font size to `28` and, under **Borders, Lines + Fills**, change the Fill color  to `#F4D481`.
8. With **StickyCheckoutButton** still selected, click on the 768 button  located in the upper-left corner of the design area. In the toolbar, change **x:** to `10` and **w:** to `748`.
9. With **StickyCheckoutButton** still selected, click on the 320 button  located in the upper-left corner of the design area. In the toolbar, change **w:** to `300`.

With **StickyDP** and **StickyCheckoutButton** placed, we are now ready to enable the **OnWindowScroll** page interaction for the **Shopping Cart** page.

Enabling the OnWindowScroll page interaction

Once the **OnWindowScroll** page interactions have been defined, the **Page Interactions** tab for the **Shopping Cart** page will look as follows:



In the sitemap, double-click on the **Shopping Cart** page to open it in the design area. Select the **Base** Adaptive View and perform the following steps:

1. To build the first case, under the design area, click on the **Page Interactions** tab.
2. Double-click on the **OnWindowScroll** event. A **Case Editor** dialog box will open.
3. Add the condition, click on the **Add Condition** button.
4. In the outlined condition box within the **Condition Builder** dialog box, perform the following steps:

    1. In the first dropdown, select **value**.
    2. In the second, enter `[[Window.ScrolY]]`.
    3. In the third dropdown, select **is greater than**.
    4. In the fourth dropdown, select **value**.
    5. In the fifth, enter `110`.
    6. Click on **OK**.

5. Create the first action to show the **StickyDP**:

    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on the **Show/Hide** drop-down menu; then, click on **Show**.
    2. Under **Configure actions**, click on the checkbox next to **StickyDP**. Next to the **More Options** dropdown, select **bring**

to front.
3. Click on **OK**.

To build the second case, under the design area, click on the **Page Interactions** tab. Double-click on the **OnWindowScroll** event. A **Case Editor** dialog box will open. In the **Case Editor** dialog, perform the following steps:

1. Create the action to hide the **StickyDP**:
    1. Under **Click to add actions**, scroll to the **Widgets** drop-down menu and click on the **Show/Hide** drop-down menu; then, click on **Hide**.
    2. Under **Configure actions**, click on the checkbox next to **StickyDP**.
    3. Click on **OK**.
2. Right-click on **Case 2** and click on **Toggle IF/ELSE IF**.

Congratulations! We have finished our e-commerce shopping cart prototype. We can now publish to AxShare and view the prototype on our mobile devices using a mobile browser or the AxShare mobile app.

Summary

In this chapter, we created an e-commerce shopping cart optimized for the iPad and iPhone by using adaptive views. We created our Base view with 1,024 width to cover the iPad landscape orientation. We also created a view at 768 width for iPad portrait and a view at 320 width for iPhone portrait.

We used a repeater widget to track items in the shopping cart. The prototype allowed us to update and remove items already in our cart. When an item was updated or removed from the repeater, the total items shown in the **SubTotalLabel** and **CartMenuItem** widgets were updated. We also updated the sub-total price shown on the **SubTotal** widget.

Index

A

- About Us page
    - design, completing / Completing the design and interactions for the Journey, Topic Detail, Charter, and About Us pages
    - interactions, defining / Defining interactions for the Journey, Topic Detail, Charter, and About Us pages, Enabling OnPageLoad actions for the Journey page
    - OnPageLoad actions, constructing / Constructing OnPageLoad actions for the About Us page
- Adaptive Views
    - about / Adaptive Views
    - adaptive view, planning / Planning for adaptive
    - managing / Managing Adaptive Views
    - URL / Managing Adaptive Views
- additional portfolio interactions
    - adding / Adding additional portfolio interactions
- aggregator
    - developing / Developing an aggregator
    - Social Content master, building / Building our Social Content master
- Article Detail page
    - designing / Designing the Article Detail page
    - OnPageLoad interaction, defining / Defining the OnPageLoad interaction

    - OnClick interactions, defining / Defining the OnClick interactions
- AxShare
    - about / Developing an aggregator
    - URL / Developing an aggregator, Publishing to AxShare
    - ebrouchure page, publishing to / Publishing to AxShare
- Axure
    - environment / The environment and interface
    - interface / The environment and interface
    - versions / The environment and interface
    - URL, for versions / The environment and interface
    - used, for content designing for eBrouchure page / Designing content with Axure for our eBrochure

B

- blog project
    - reviewing / Reviewing our project
    - background information / Background information for our blog
    - planning / Planning our blog
    - sitemap, inspecting / Inspecting our sitemap, Reviewing the registration and post flows
    - registration, reviewing / Reviewing the registration and post flows
    - post flows, reviewing / Reviewing the registration and post flows
- BreadCrumb master
    - constructing / Constructing our BreadCrumb master
    - Repeater widget, formatting / Formatting the Repeater widget and placing widgets on the BreadCrumb master

    - widgets, placing / Formatting the Repeater widget and placing widgets on the BreadCrumb master
    - Secondary_Page_Body master, assembling / Assembling our Secondary_Page_Body master
    - widgets, placing on Secondary_Page_Body master / Placing widgets on the Secondary_Page_Body master
    - Informational_Page_Body master, manufacturing / Manufacturing our Informational_Page_Body master
- BreadCrumb repeater
    - OnItemLoad event, updating / Updating the Repeater item's OnItemLoad event for the BreadCrumb repeater
    - OnItemLoad event actions, defining / Defining the OnItemLoad event actions

C

- Cart repeater interactions
    - defining / Defining the Cart repeater interactions
    - OnSelectionChange case of Quantity widget, creating / Creating the Quantity widget's OnSelectionChange case
    - OnClick cases of DeleteButton widget, creating / Creating the DeleteButton widget's OnClick cases
    - Cart repeater OnItemLoad event cases, creating / Creating cases for the Cart repeater OnItemLoad event
- Charter page
    - design, completing / Completing the design and interactions for the Journey, Topic Detail, Charter, and About Us pages
    - interactions, defining / Enabling OnPageLoad actions for the Journey page
    - OnPageLoad actions, facilitating / Activating OnPageLoad actions for the Topic Detail page, Facilitating OnPageLoad actions for the Charter page

- child page
    - creating / Creating a child page
- commentary creation
    - enabling / Enabling commentary and new post creation
    - facilitating / Facilitating commentary creation
- community site
    - reviewing / Reviewing our project
    - client, background information / Background information for our client
    - architecting / Architecting our community site
    - sitemap, inspecting / Inspecting our sitemap
    - forum, reviewing / Reviewing the forum and news flow
    - news flow, reviewing / Reviewing the forum and news flow
    - masters, building / Identifying and building our masters
    - masters, identifying / Identifying and building our masters
    - page design, finalizing / Finalizing page design for our community site
- content designing, for eBrouchure page
    - Axure, using / Designing content with Axure for our eBrochure
    - cover page, designing / Designing our cover page
    - page 1, designing / Designing page1
    - page 2, designing / Designing page2
    - page 3, designing with panoramic image / Designing page3 using the left-hand side of our panoramic image
    - page 4, designing with panoramic image / Designing page4 using the right-hand side of our panoramic image
    - back page, designing / Designing our back page

D

- design area
    - about / The environment and interface, The design area
    - panes / The environment and interface
    - pages, opening / Opening pages
    - grid, displaying / Showing the grid
    - page guides, adding / Adding page guides
- Dropbox
    - URL / Converting pages into images
- dynamic panel
    - URL / Enhancing our Header master

E

- e-commerce shopping cart
    - designing / Designing an e-commerce shopping cart
    - sitemap, updating / Updating the sitemap and configuring Adaptive Views
    - Adaptive Views, configuring / Updating the sitemap and configuring Adaptive Views
    - global variables, creating / Creating global variables
    - masters, adding to Master pane / Adding masters to the Masters pane
    - Header master, designing / Designing the Header master
    - Footer master, designing / Designing the Footer master
    - page, designing / Designing the Shopping Cart page
- e-commerce shopping cart page
    - designing / Designing the Shopping Cart page
    - Cart repeater, designing / Designing the Cart repeater
    - Cart repeater, updating / Updating the Cart repeater for the 768 and 320 views

    - Cart repeater interactions, defining / Defining the Cart repeater interactions
    - sticky Proceed, creating to Checkout button / Creating the sticky Proceed to checkout button
    - OnWindowScroll page Interaction, enabling / Enabling the OnWindowScroll page interaction
- eBrouchure page
    - creating / Creating our eBrochure page
    - content, designing with Axure / Designing content with Axure for our eBrochure
- eMagazine
    - designing / Designing an eMagazine
    - sitemap, updating / Updating the sitemap and configuring Adaptive Views
    - Adaptive Views, configuring / Updating the sitemap and configuring Adaptive Views
    - Global Variables, creating / Creating global variables
    - masters, adding to Masters pane / Adding masters to the Masters pane
    - Header master, designing / Designing the Header master
    - Footer master, designing / Designing the Footer master
    - home page, designing / Designing the eMagazine home page
    - Topic Overview page, designing / Designing the Topic Overview page
    - Article Detail page / Designing the Article Detail page
- Enter Now flow, photo contest
    - creating / Creating the Enter Now flow
    - Enter page, designing / Designing the Enter page
    - Upload Photo page, designing / Designing the Upload Photo page
- Entry Confirmation page, photo contest

    - designing / Designing the Entry Confirmation page

F

- Facebook brand assets
    - reference link / Designing the Header master
- Facebook feed
    - adding / Adding Facebook and Twitter to our feed
- Footer master
    - forming / Forming our Footer master
    - widgets, placing / Placing widgets on the Footer master
    - interactions, building / Building interactions for our Footer master
    - interactions, deleting for Label widgets / Defining interactions for our Label widgets
- Footer master, photo contest
    - designing / Designing the Footer master
- Forum master
    - fabricating / Fabricating our Forum master
    - Repeater, formatting / Formatting the Repeater and placing widgets on the Forum master
    - widgets, placing / Formatting the Repeater and placing widgets on the Forum master
    - interactions, building / Building interactions for the Forum master
    - OnClick interactions, enabling for Category Hot Spot / Enabling the OnClick interactions for the Category Hot Spot
    - OnClick interactions, activating for Topic Hot Spot / Activating the OnClick interactions for the Topic Hot Spot

- Forum page
    - design and interactions, manufacturing / Manufacturing design and interactions for the Forum page
    - OnPageLoad actions, defining / Defining OnPageLoad actions for the Forum page
- Forum repeater
    - OnItemLoad event, updating / Updating the Repeater item's OnItemLoad event for the Forum repeater

G

- Gallery page, photo contest
    - designing / Designing the Gallery page
    - Photo Gallery repeater, designing / Designing the PhotoGallery repeater
    - Photo Gallery repeater, updating / Updating the PhotoGallery repeater for the 768 and 320 views
    - Photo Gallery repeater interactions, defining / Defining PhotoGallery repeater interactions
    - sorting interactions, defining / Defining sorting and pagination interactions
    - paging interactions, defining / Defining sorting and pagination interactions
    - sorting interactions, defining for SortDroplist / Enabling sorting for SortDroplist
    - paging controls, enabling / Enabling pagination controls
- grid
    - displaying / Showing the grid

H

- Header master
    - designing / Designing our Header master
    - widgets, placing / Placing widgets on the Header master
    - interactions, building / Building interactions for our Header master
    - enhancing / Enhancing our Header master
    - interactions, refining / Refining the interactions for our Header master
    - interactions, defining for LoginButton / Defining interactions for the LoginButton
    - interactions, enabling for NewUserLink / Enabling interaction for the NewUserLink
    - interactions, activating for ForgotLink / Activating interactions for the ForgotLink
    - interactions, facilitating for LogOutButton / Facilitating interactions for the LogOutButton
    - interactions, constructing for RegistrationLightBoxDP / Constructing interactions for the RegistrationLightBoxDP
- Header master, photo contest
    - designing / Designing the Header master
- Help page
    - design and interactions, completing / Completing the design and interactions for the Terms, Privacy, and Help pages, Creating the Initialize case for the OnPageLoad event for the Terms page
- home page, eMagazine
    - designing / Designing the eMagazine home page

- images
    - pages, converting to / Converting pages into images
- Informational_Page_Body master
    - manufacturing / Manufacturing our Informational_Page_Body master
    - widgets, placing / Placing widgets on the Informational_Page_Body master
- interactions
    - refining, for Header master / Refining the interactions for our Header master
    - refining, for LoginButton / Defining interactions for the LoginButton
    - user's email, validating / Validating the user's email and password
    - user's password, validating / Validating the user's email and password
    - ShowErrorMessage case, creating / Creating the ShowErrorMessage case
    - enabling, for NewUserLink / Enabling interaction for the NewUserLink
    - activating, for ForgotLink / Activating interactions for the ForgotLink
    - facilitating, for LogOutButton / Facilitating interactions for the LogOutButton
    - RegistrationLightBoxDP, hiding / Updating Variables and hiding the RegistrationLightBoxDP
    - variables, updating / Updating Variables and hiding the RegistrationLightBoxDP
    - ShowLoginState Case, creating / Creating the ShowLoginState case
- interactions, building for Header master

    - about / Building interactions for our Header master
    - defining, for CommunityLogo / Defining interaction for CommunityLogo
    - enabling, for HzMenu / Enabling interactions for HzMenu
    - illuminating, for EmailTextField / Illuminating interactions for EmailTextField
    - constructing, for PasswordTextField / Constructing interactions for the PasswordTextField
- interactions, constructing for PasswordTextField
    - about / Constructing interactions for the PasswordTextField
    - OnMouseEnter event, activating / Activating the OnMouseEnter event
    - OnMouseOut Event, facilitating / Facilitating the OnMouseOut Event
- interactions, illuminating for EmailTextField
    - about / Illuminating interactions for EmailTextField
    - OnMouseEnter event, activating / Activating the OnMouseEnter event
    - OnMouseOut event, facilitating / Facilitating the OnMouseOut event
- interface
    - about / The environment and interface
    - main menu / The environment and interface
    - toolbar / The environment and interface
    - design area / The environment and interface, The design area
    - pages / Pages
    - panes / Panes

J

- Journey page

    - interactions, defining / Defining interactions for the Journey, Topic Detail, Charter, and About Us pages, Enabling OnPageLoad actions for the Journey page
    - OnPageLoad actions, enabling / Enabling OnPageLoad actions for the Journey page
- jPageFlipper 0.9
    - URL / Converting pages into images

M

- main menu
    - about / The environment and interface
- masters
    - about / Architecting our community site
    - building / Identifying and building our masters, Updating and building our masters
    - identifying / Identifying and building our masters
    - global variables, creating / Creating global variables
    - adding, to Masters pane / Adding masters to the Masters pane, Adding masters to the Masters pane
    - Header master, designing / Designing our Header master
    - BreadCrumb master, constructing / Constructing our BreadCrumb master
    - Footer master, forming / Forming our Footer master
    - updating / Updating and building our masters
    - additional global variables, creating / Creating additional global variables
    - Header master, enhancing / Enhancing our Header master
    - data, refreshing from Forum repeater / Refreshing data for our Forum repeater
    - Post Commentary master, assembling / Assembling our Post Commentary master

    - Post Commentary master, assesmbling / Assembling our Post Commentary master
    - NewPost master, manufacturing / Manufacturing a NewPost master
- Masters pane
    - about / Masters
    - masters, adding to / Adding masters to the Masters pane

N

- new post creation
    - enabling / Enabling commentary and new post creation
    - allowing, for authorized users / Allowing new post creation to authorized users

O

- OnItemLoad event
    - updating, for Forum repeater / Updating the Repeater item's OnItemLoad event for the Forum repeater
    - actions, realizing / Realizing OnItemLoad event actions
    - UpdateTopic case, defining / Defining the UpdateTopic case
    - UpdateCategory case, defining / Defining the UpdateCategory case
- OnWindowScroll cases
    - WindowScroll Y < 400 case, defining / Defining the WindowScroll Y < 400 case
    - WindowScroll Y > 400 & Y < 800 case, enabling / Enabling the WindowScroll Y > 400
    - WindowScroll Y > 800 case, creating / Creating the WindowScroll Y > 800 case

- OnWindowScroll interactions
    - enabling / Enabling OnWindowScroll interactions
    - WindowScrolled case, defining / Defining the WindowScrolled case
    - remaining OnWindowScroll cases, completing / Completing the remaining OnWindowScroll cases

P

- page body, community site
    - designing / Designing the page body
    - widgets, placing in carousel / Placing widgets and interactions for the dynamic carousel
    - interactions, building for carousel dynamic panel / Building interactions for the carousel dynamic panel, Enabling interactions for the CheckForClick Hot Spot, Defining the OnClick interactions for the Carousel dynamic panel
    - interactions, enabling for CheckForClick Hot Spot / Enabling interactions for the CheckForClick Hot Spot
    - OnClick interactions, defining for Carousel dynamic panel / Defining the OnClick interactions for the Carousel dynamic panel
    - Previous and Next buttons, activating / Activating the Previous and Next buttons
    - widget, placing / Placing widgets and adding interactions for calls to action
    - interactions for calls to action, adding / Placing widgets and adding interactions for calls to action
    - social media feed / Adding a social media feed
- page design, community site
    - finalizing / Finalizing page design for our community site
    - widgets, placing on Home page / Placing widgets and masters on the Home page

    - masters, placing on Home page / Placing widgets and masters on the Home page
    - page body, designing / Designing the page body
    - masters, adding / Adding masters and defining actions for the home page
    - actions for home page, defining / Adding masters and defining actions for the home page
    - for About Us page, completing / Completing the design and interactions for the Journey, Topic Detail, Charter, and About Us pages
    - for Journey page, completing / Completing the design and interactions for the Journey, Topic Detail, Charter, and About Us pages
    - for Topic Detail page, completing / Completing the design and interactions for the Journey, Topic Detail, Charter, and About Us pages
    - for Charter page, completing / Completing the design and interactions for the Journey, Topic Detail, Charter, and About Us pages
- page flipper AxShare plugin
    - creating / Creating a page flipper AxShare plugin
    - publishing, to AxShare / Publishing to AxShare
- page interactions, updating
    - about / Updating page interactions
    - OnPageLoad event, defining / Defining the OnPageLoad event
    - OnWindowScroll interactions, enabling / Enabling OnWindowScroll interactions
- Page Properties pane
    - about / The Page Properties pane
- pages

    - about / Pages
    - Sitemap pane / The Sitemap pane
    - child page, creating / Creating a child page
    - opening / Opening pages
    - page guides, adding / Adding page guides
    - converting, to images / Converting pages into images
- page style
    - reference link / The Page Properties pane, Designing our parallax site
- panes
    - about / Panes
    - Widgets pane / The widgets pane and libraries
    - libraries / The widgets pane and libraries
    - Masters pane / Masters
    - Page Properties pane / The Page Properties pane
    - Widget Interaction and Notes pane / The Widget Interaction and Notes pane
    - Widget Properties and Style pane / The Widget Properties and Style pane
    - Widget Manager pane / The Widget Manager pane
- parallax site, designing
    - about / Designing our parallax site
    - page anchors, placing / Placing page anchors
    - background dynamic panel, building / Building the background dynamic panel
    - sectional content, creating / Creating our sectional content
- photo contest, designing
    - about / Designing a photo contest
    - sitemap, updating / Updating the sitemap and configuring Adaptive Views

    - Adaptive Views, configuring / Updating the sitemap and configuring Adaptive Views
    - global variables, creating / Creating global variables
    - masters, adding to Masters pane / Adding masters to the Masters pane
    - Header master, designing / Designing the Header master
    - Footer master, designing / Designing the Footer master
    - Photo Contest home page, designing / Designing the Photo Contest home page
    - Enter Now flow, creating / Creating the Enter Now flow
    - Entry Confirmation page, designing / Designing the Entry Confirmation page
    - Gallery page, designing / Designing the Gallery page
    - View Entry page, designing / Designing the View Entry page
    - Prize page, designing / Designing the Prizes page
    - Rules page, designing / Designing the Rules page
- Pinterest board
    - including / Including a Pinterest board
    - URL / Including a Pinterest board
- Post Commentary master
    - assembling / Assembling our Post Commentary master
    - OnItemLoad event actions, defining / Defining the OnItemLoad event actions
- Privacy page
    - design and interactions, completing / Completing the design and interactions for the Terms, Privacy, and Help pages, Creating the Initialize case for the OnPageLoad event for the Terms page
    - initial case, for OnPageLoad event, creating / Creating the Initialize case for the OnPageLoad event for the Privacy page, Creating the Initialize case for the OnPageLoad event for the Help page

- Prizes page, photo contest
    - designing / Designing the Prizes page

R

- Rules page, photo contest
    - designing / Designing the Rules page

S

- sitemap
    - pages, repurposing / Repurposing pages in our sitemap
    - home page, remodeling / Remodeling our home page
    - remaining pages, invigorating / Invigorating our remaining pages

  / Designing the Footer master
- Sitemap pane
    - about / The Sitemap pane
- SnapWidget Instagram widget
    - URL / Pulling Instagram content
- Social Content Aggregator
    - about / Developing an aggregator
- Social Content master
    - building / Building our Social Content master
    - Facebook, adding to feed / Adding Facebook and Twitter to our feed

    - Twitter, adding to feed / Adding Facebook and Twitter to our feed
    - Instagram content, pulling / Pulling Instagram content
    - Pinterest board, including / Including a Pinterest board

T

- tab control
    - URL, for tutorial / Adding additional portfolio interactions
- Terms page
    - design and interactions, completing / Completing the design and interactions for the Terms, Privacy, and Help pages, Creating the Initialize case for the OnPageLoad event for the Terms page
    - initial case, for OnPageLoad event, creating / Creating the Initialize case for the OnPageLoad event for the Terms page
- Tint / Adding a social media feed
- Tint social aggregator
    - URL / Adding Facebook and Twitter to our feed
- toolbar
    - top toolbar / The environment and interface
    - bottom toolbar / The environment and interface
- Topic Detail page
    - design, completing / Completing the design and interactions for the Journey, Topic Detail, Charter, and About Us pages
    - interactions, defining / Defining interactions for the Journey, Topic Detail, Charter, and About Us pages, Enabling OnPageLoad actions for the Journey page

    - OnPageLoad actions, activating / Activating OnPageLoad actions for the Topic Detail page
- Topic List page
    - design and interactions, fabricating / Fabricating design and interactions for the Topic List page
    - OnPageLoad actions, defining / Defining OnPageLoad actions for the Topic List page
    - initialize case, creating for OnPageLoad event / Creating the Initialize case for the OnPageLoad event
    - SetAllTopicFilter case, adding for SetAllTopicFilter case / Adding the SetAllTopicFilter case for the OnPageLoad event
- Topic Overview page
    - designing / Designing the Topic Overview page
    - OnPageLoad interaction, defining / Defining the OnPageLoad interaction
    - OnClick interactions, defining / Defining OnClick interactions
- Twitter brand assets
    - reference link / Designing the Header master
- Twitter feed
    - adding / Adding Facebook and Twitter to our feed

V

- View Entry page, photo contest
    - designing / Designing the View Entry page

W

- wayfinding / Constructing our BreadCrumb master
- Widget Interaction and Notes pane
    - about / The Widget Interaction and Notes pane
- Widget Manager pane
    - about / The Widget Manager pane
- Widget Properties and Style pane
    - about / The Widget Properties and Style pane
- widgets
    - reference link / The Widget Properties and Style pane
- Widgets pane
    - about / The widgets pane and libraries