Cascading Style Sheets

Cascading Style Sheets (CSS) define how HTML elements are displayed. Styling can happen in a few different places. The most common place for CSS is an external style sheet (which is a file with a .css extension). You place a <link> between your HTML <head> tags like Listing A–1.

Listing A–1. HTML header - External Stylesheet

```
<head>
    <link href="stylesheet.css" rel="stylesheet" type="text/css">
</head>
```

**NOTE:** You can use relative or absolute paths to your style sheets for the href attribute of <link>.

You can also place a <style> tag in the <head> portion of your HTML document and define your CSS there (Listing A–2); this is called an internal style sheet.

Listing A–2. HTML Header - Internal Stylesheet

```
<head>
    <style type="text/css">
        ...
    </style>
</head>
```

Finally you can add a style attribute to any HTML element and define your styles there; this is called an inline style (Listing A–3).

Listing A–3. Inline Style

```
<div style="width:50px;height:50px;">...</div>
```

The Cascading in Style Sheets

When an HTML element has multiple styles defined on it, the one with the highest priority will be chosen and override the rest. An inline style (a style defined on the HTML element) has the highest priority and will override any other CSS defined. Next is an internal style sheet (one defined in the header of your HTML document), then external
style sheets (you reference these with a link tag in the header of your HTML document, which are typically declared before the internal style sheets). Finally, browser default options are at the bottom of the list, and will have the lowest priority.

- Inline styles
- Internal/external style sheets (last one defined determines style)
- Browser defaults

Note that overwriting happens only if the specificity of the selectors is the same. So, for example, let’s say you have a style that applies to p elements in a div, and then later on you have a style that applies to all p elements.

p elements inside a div will get the first style, because the most specific style wins, even if there’s a more general one later.

Style declarations aren’t monolithic. When something gets "overridden," what’s really happening is that any declarations that are the same level of specificity and the same property are overridden, but all the other properties remain.

So, for example, let’s say you have something similar to Listing A–4.

**Listing A–4. Paragraph tag with color**

```
<
  div p { /* applies to p elements inside a div */
    color: blue;
  }
```

And then later on, you have Listing A–5.

**Listing A–5. Paragraph Tag with color and text decoration**

```
  p { /* applies to all p elements */
    color: black;
    text-decoration: underline;
  }
```

<p> elements in a div will be blue and underlined and all other <p> elements will be black. The more-specific declaration has a color so that overrides the general color, even though it is defined first. Because it doesn’t say anything about text-decoration, that style is determined from the more-general set.

**CSS Syntax**

A typical CSS statement looks like this:

```
SELECTOR {DECLARATION[PROPERTY: VALUE];DECLARATION[PROPERTY:VALUE]; }
```

For example, consider Listing A–6.
Listing A–6. Header 1 tag with color

h1 { color: #FFFFFF; }

Selector-> h1
Declaration-> color: #FFFFFF;
Property-> color
Value-> #FFFFFF

CSS declarations always end with a semicolon, and curly brackets surround declaration groups.

NOTE: Do not leave spaces between property values and units.

Incorrect  top: 20 px;
Correct top: 20px;

Comments

A CSS comment begins with "/*", and ends with "*/", like Listing A–7.

Listing A–7. Comments

/* This is a comment */

/*
   This is a
   multiline
   comment
*/

Identifying Elements with ID and Class

ID defines a special and unique case for an element (this means that it can only be used once per document). These should be treated like global variables and used sparingly. In CSS, an ID is declared with a pound sign (#) followed by a unique name, such as #unique_box in Listing A–8.

Listing A–8. CSS ID Example

<html>
  <head>
    <style type="text/css">
      #unique_box {
        width: 50px;
        height: 50px;
        background-color: blue;
      }
    </style>
  </head>
  <body>
    <div id="unique_box"></div>
  </body>
</html>
In CSS, if you follow a class declaration with a selector, you can define specific declarations for that element.

CSS classes define a special non-unique case for elements. Classes should be used when multiple elements require the same styling. CSS classes are declared with a period (.) followed by a unique name, such as .box in Listing A–9.

Listing A–9. CSS Class Example

```html
<html>
<head>
  <style type="text/css">
    .box {
      width: 50px;
      height: 50px;
    }
  </style>
</head>
<body>
  <div class="box"></div>
  <div class="box"></div>
  <div class="box"></div>
</body>
</html>
```

In CSS, if you follow a class declaration with a selector, you can define specific declarations for that element. Where ".box" is the class, "p" is the selector, and "color: green;" is the declaration (Listing A–10).

Listing A–10. Apply a class to a <p> tag

```css
.box p {
  color: green;
}
```

Common Patterns

Generally, you won’t be writing CSS that applies to all <p> elements, or all <a> elements. You will write CSS that applies only to certain elements based on how they are placed relative to other elements. For example, you might have a specific style for all <p> elements inside any <div> with class 'bounding-box.'

Examples of nesting selectors include Listing A–11 and examples of grouping selectors include listing A–12.

Listing A–11. Examples of nesting selectors

```css
div p { /* all p elements that are inside a div */
  color: green;
}
div p.box { /* all p elements with class box that are inside a div */
  color: black;
}
Listing A–12. Examples of grouping selectors

/* all p and h1 elements inside the div with class main-text */
div.main-text p, div.main-text h1 {
  color: black;
}

Common CSS Attributes (Display: block versus inline)

The display property controls how an element is displayed. It does this with two properties called block and inline. The block property tells the element to take up the full width available and forces line breaks in text. The inline property tells the element to take up just as much width as necessary and doesn’t force line breaks.

NOTE: "display: none;" will hide an element, making it invisible.

These HTML elements have a display: block; by default:
<p>, <h1>…<h4>, <div>

These HTML elements have a display: inline; by default:
<a>, <span>

Visibility has two values, visible or hidden, to control whether an element is visible or not. [visibility: hidden;]

Margin clears the area outside of the container. Margin takes four values in a clockwise rotation: MARGIN TOP RIGHT BOTTOM LEFT. Each value must be defined in pixels, pt, em, or % (Listing A–13).

NOTE: Negatives values are allowed, so that you may overlap content.

Listing A–13. Margin Example

margin-left: VALUE;
margin-right: VALUE;
margin-top: VALUE;
margin-bottom: VALUE;

Padding clears the area inside the container (Listing A–14). Padding takes four values in a clockwise rotation: PADDING TOP RIGHT BOTTOM LEFT. Each value must be defined in pixels, pt, em, or %. [ p { padding: 0px 10px 0px 10px;} ]
NOTE: Negative values are not allowed.

Listing A–14. Padding Example

```css
padding-left: VALUE;
padding-right: VALUE;
padding-top: VALUE;
padding-bottom: VALUE;
```

Background controls the background color or image of an HTML element (Listing A–15). It has options BACKGROUND: COLOR IMAGE REPEAT ATTACHMENT POSITION.

```css
body { background: #00ff00 url('image.png') no-repeat fixed top; }
```

Listing A–15. Background Example

```css
background-color: VALUE;
background-image: VALUE;
background-repeat: VALUE;
background-attachment: VALUE;
background-position: VALUE;
```

Color controls text color. Colors can be defined by name [color: red;], RGB [color: rgb(255,0,0);], or hex representation. [color: #ff0000;].

Text-align is used to set the horizontal alignment of text. [p {text-align: center;}]

Text-decoration allows you to over-line, under-line, line-through, or blink text. The blink option will flash the text and hide it at a fixed rate. It is not supported in IE, Safari, or Chrome. It is most commonly used to remove the decoration for link elements. [a { text-decoration: none; }]

Text-transform is used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word (Listing A–16). [h1 { text-transform: uppercase; }]

Float specifies how elements lay out relative to each other. Elements can be told to move as far left or right as they can, allowing other elements to wrap around them. Floating <div>(s) or <img>(s) is common.

Listing A–16. Float Example

```html
<html>
<head>
  <style type="text/css">
    img {
      float: right;
    }
  </style>
</head>
<body>
<p> This text is only here to show wrapping around the image. You will see that the text will continue to flow on the left around the image on the right. You will also see that the image has floated as far right as possible (Listing A–17).
```
Listing A–17. Text wrapping and images

```html
<img src="image.jpg" width="50" height="50" alt="some image" />
</body>
</html>
```

**TIP:** Elements after the floated element(s) will continue to wrap. To avoid this, use the clear property on the elements you do not want floated (Listing A–18). Values of clear are: left, right, both, none, inherit.

Listing A–18. Clear Example

```css
.foo {
  clear: both;
}
```
Index

A

<a> tag, 190–191
Action menu, File System Editor, 78
Activate for BlackBerry option, 137
ActiveSync window, Visual Studio 2008, 79
adb devices utility, 49
Add a New Item context menu item, Visual Studio 2008, 75
Add Devices button, iPhone Developer Program Portal, 30
Add New Project dialog box, Visual Studio 2008, 77
Add New Smart Device Project wizard, Visual Studio 2008, 67–68
Add or Remove Programs screen, Visual Studio 2008, 77
Add Project Output Group dialog box, File System Editor, 78
addGlossToIcon option, 216
AdHoc application, 31
ADT (Android Development Tools), 36, 136
ADT Layout Editor, 41
ADT plug-in, 41
Advanced panel, 229
Advanced tab, 212
Ajax, adding screens with jQTouch, 212–214
anchor tag, 218
Android
   building for Android device, 48–49
development for
      building application, 39–46
      with Eclipse, 36–38
      embedding WebView in application, 46–48
digitally signing application, 50
HTML and CSS support on, 164
overview, 35
PhoneGap, open source framework for, 136
and Rhodes
debugging on, 101
running application on, 94
Android Development Tools (ADT), 36, 136
Android Market, 50
Android SDK and AVD Manager, 37
Android Virtual Device (AVD), 37
Android WebKit browser control, 170
/android/framework/assets/www directory, 140
Any iPhone OS Device option, Xcode, 32–33
.apk file, 50
app folder, 90, 96–97
App IDs, iPhone Developer Program Portal, 31
App Store style, list styles with iWebKit, 191–192
App World, BlackBerry, 63–64
Appearance section, Visual Studio 2008, 70, 73
app.js file, 155, 158
Apple Mail application, 220
Application Folder, 78
Application ID field, Titanium, 154
Application tab, 49
application.css file, 226
application.js file, 227
application.rb file, 91, 121, 129
applications
   adding to CAB Projects, 78
   building and testing, 58
   creating shortcuts, 78
distributing, 80
   embedding Web View in, 75
   layout template, adding iWebKit framework to, 204
   marketplace, 2–4
applist class, 187
app/Product/index.erb page, 97
apps
   building. See also base functionality; interfaces, creating
Index

adding WebBrowser controls, 75
create Xcode project, 19
creating HTML pages, 75
creating Smart Device Projects, 67
deploying and testing, 72
embedding Web View in applications, 75
loading HTML in WebBrowser controls, 76
overview, 18
installing on devices
creating provisioning profiles, 32
finding device IDs, 31
install provisioning profiles, 32
installing and running on devices, 32–33
manually setting up iPhone provisioning, 30
using development provisioning assistant, 29–30
packaging and distributing
adding applications to CAB Projects, 78
adding CAB Projects to solutions, 77
adding Registry entries, 78
building and deploying CAB files, 78–79
creating application shortcuts, 78
customizing product names, 77
installing CAB files, 79
overview, 76
app/Settings/setting.yml file, 116
Appstore application, 200
arrow class, 220
Ask, Julie, 4
Associated Attributes for Backend
Associated Attributes for Backend
Attributes Inspector, Interface Builder, 23(attributes, for CSS, 251–253
Attributes Inspector, Interface Builder, 28
authenticate method, in RhoSync, 121
authentication, in RhoSync, 116–117
Authenticode Signature option, Solution Explorer, 79
auto class, 194
AVD (Android Virtual Device), 37

background
attribute, 252
image property, 173
position attributes, 176
backSelector option, 216
base functionality
adding buttons to views, 69
creating click event handlers, 71–72
customizing buttons, 70
overview, 68
basic views, in jQTouch, 217–218
BasicEditField class, 60
/beatles HREF, 214
/beatles link, 213
Behavior section, Visual Studio 2008, 76
Berries button, 86
BES (BlackBerry Enterprise Server), 52
Bigfield class, 197
BlackBerry
applications, building and testing, 58
BlackBerry Browser Field
BlackBerry App World, 63–64
building for BlackBerry devices, 62–63
over the air (OTA) distribution, 63
overview, 61
browser-rendering engine 4.2
CSS in, 237–239
dynamic layout in with Rhodes, 242–243
fonts for, 239–241
frames in, 241
JavaScript support in, 241
browser-rendering engine 4.6, 244
code, 57–60
creating Eclipse project, 53–55
creating interfaces, 55–57
development environment for, 245
HTML and CSS support on, 165
Java Development, 52–53
labels, text fields, and buttons, 58–60
and limitations of target browser, 236
overview, 235
PhoneGap, open source framework for, 137
platform of, 51–52
and Rhodes
debugging on, 101
running application on, 94–95
screen resolutions for, 244–245
user interaction on, 244–245

B
Back button, 214–215, 218
back class, 214
BlackBerry Enterprise Server (BES), 52
BlackBerry JDE Plug-in, 137
BlackBerry menu, Eclipse, 55
BlackBerry Project option, Eclipse, 54
BlackBerry Signature Tool, 62
BlackBerry Workspace, Eclipse, 55
blackberry/framework/ directory, 137, 140
Blue Button - Left element, 195
Blue Button - Right element, 195
body, with iWebKit, 186
<body> tag, 186–187, 190, 194, 204, 216
browser.field (4.2) engine, 236
BrowserField class, 61
browser-rendering engine 4.2, for BlackBerry
CSS in, 237–239
dynamic layout in with Rhodes, 242–243
fonts for, 239–241
frames in, 241
JavaScript support in, 241
browser-rendering engine 4.6, for BlackBerry, 244
Build and Go option, Xcode, 26, 33
Build and Run as Android Application menu option, 136
Build and Run command, Xcode, 157
Build and Run option, 135, 142, 205
Build page, Solution Explorer, 79
Build tab, Xcode, 33
Build Target list, 40
build.yml file, 90, 93
Button Bar, Android, 182
button bars, HTML and CSS support for, 168
Button class, 44, 215
Button control, 44
button xtype, 230
buttons
  adding to views, 69
  customizing, 70
  layout of, 22–23
  overview, 58–60
  on screens with jQTouch, 215

C
CAB files, 78–79
CAB Projects
  adding applications to, 78
  adding to solutions, 77
CABProject Property Pages dialog box, Solution Explorer, 77
CABProject\Debug folder, 79
cacheGetRequests option, 216
Camera example
  in PhoneGap, 150–152
  Rhodes application framework, 106–108
  Titanium Mobile, 158–160
cancel and back, adding screens with jQTouch, 214–215
cancel class, 214
Cascading Style Sheets. See CSS
Certificate box, Solution Explorer, 79
Certificates option, iPhone Developer Program Portal, 30
check boxes, HTML and CSS support for, 169–171
checkbox class, 198, 230
Choose button, 135
class, identifying elements in CSS with, 250
class attribute, 209, 220
Class menu item, Eclipse, 55
classic style, list styles with iWebKit, 188–190
Classic type, 187
clean.bat file, 101
click event handlers, creating, 71–72
Click Select from Store button, Solution Explorer, 79
close/back action, 111
code, 25, 57–60
Code Signing Identity, Xcode, 33
color attribute, 252
comments, in CSS, 249
components
  adding in Sencha Touch, 231
  xtype, 230
com.xplatform.helloworld package, 55
Configure BlackBerry Workspace, Eclipse, 55
classifying code to views, 26
Connection Inspector, Interface Builder, 28
contact_controller.rb file, 104
Contact/index.erb page, 105
Contacts example
  in PhoneGap, 146–150
  Rhodes application framework, 103–106
Contacts/app/Contact/index.erb page, 105
Contacts/app/Photo/index.erb page, 107
Contacts/app/Photo/photo_controller.rb file, 107
container xtype, 230
ccontent <div> tag, 188
content class, 186
ContentView control, 47
ccontext menus, HTML and CSS support for, 168
ccontroller code, writing, 23–25
ccontrols, WebBrowser, 75–76
Copy items into destination group’s folder (if needed) check box, 201
Copy to Output Directory field, solution browser, 75
ccreate, read, update, and delete (CRUD), 96
Create Application button, RhoHub, 123
Create Folder References for any added folders check box, 201
Create Folder References option, 201
ccreate method, in RhoSync, 119
Create New Object page, RhoHub, 123
Create Object button, RhoHub, 123
ccreate_hash argument, 119
cross-platform development, 9–10
cross-platform frameworks, 5–6, 10–13
CRUD (create, read, update, and delete), 96
CSS (Cascading Style Sheets)
capplying to elements, based on placement, 250
ncattributes for, 251–253
on BlackBerry, browser-rendering engine 4.2, 237–239
nccomments in, 249
ncidentifying elements, 249–250
npriority of styles in, 247–248
nsyntax for, 248–249
css folder, 227
cubeSelector option, 216
current_user method, 116
current_user.login method, 116

cDashboard, RhoHub, 125
data storage format, in RhoSync, 114
database (Rhom), for Rhodes, 86–87
datapanel xtype, 230
dataview xtype, 230
dealloc method, 25
debug\HelloWorld.cab, 77
debug-with-comments.js file, 226
delete method, in RhoSync, 120
demo folder, 201

cDesign section, Visual Studio 2008, 70
cDesign view, Visual Studio 2008, 68
cDeveloper menu, 229
cDeveloper Portals Device registration page, 31
cdevelopment
architecture, for Rhodes, 84–85
environment, for BlackBerry, 245
provisioning assistant, 29–30
device Application template, Visual Studio 2008, 67
device capabilities, in Titanium Mobile, 157–158
Device Chooser dialog box, 49
Device Emulator Manager menu item, Visual Studio 2008, 76, 79
devices
 capabilities with Rhodes, 101–102
IDs, finding, 31
iPhone Developer Program Portal, 30
digitally signing application for Android, 50
Directory field, Titanium, 154
display: none attribute, 251
dissolveSelector option, 216
<div> tag, 186, 194, 250
dOM (Document Object Model), 207, 236
doublead class, 191
downloads folder, 58
downward-facing disclosure indicator, 173
dump_pic callback, 152

EEclipse
creating Eclipse project, 53–55
development for Android with, 36–38
edgetoedge class, 218
Edit Text item, 43
Editor screen, RhoHub, 126
Editor tab, RhoHub, 126
EditText control, 45
EditText item, 42–43
elements, in CSS, 249–250
Embedded Ruby (ERB), 84, 242
enterEventDispatcher( ) method, 57
Entity-Attribute-Value (EVA), 114
ERB (Embedded Ruby), 84, 242
EVA (Entity-Attribute-Value), 114
event handlers
click, 71
creating, 72
example applications
   Rhodes application framework
      Camera, 106–108
      Contacts, 103–106
      Geolocation, 108–111
      RhoSync, product inventory, 122–130
      Titanium Mobile, Camera, 158–160
Ext.getCmp('mainscreen') call, 232
Ext.setup method, 227
Ext.TabPanel class, 232
ext-touch.css file, 226
ext-touch-debug.js file, 226
ext-touch.js file, 226

F
fadeSelector option, 216
fail callback, 149
field xtype, 230
fieldChanged method, 60
FieldChangeListener method, 60
fieldset xtype, 230
<form> tag, 196, 199
Field menu
   Registry Editor, 78
   Visual Studio 2008, 67, 77
File System context menu item, Solution Explorer, 78
File System Editor, 78
File System on Target Machine, File System Editor, 78
files, CAB, 78–79
fixedViewport option, 216
flipSelector option, 216
float attribute, 252
fonts, on BlackBerry, 239–241
Force https://... option, Android SDK and AVD Manager, 37
Force https://... sources to be fetched using http://..., check box, 109
form xtype, 230–231
<form> tag, 196
Form1.cs Design view, Visual Studio 2008, 73
Form1.cs file, 71
forms, with iWebKit, 196–199
formSelector option, 216
Forward button, 215
)frame> element, 241
frames, on BlackBerry, 241
<formset> element, 241

Framework folder, 204
frameworks, cross-platform, 5–6, 10–13
fullscreen (Boolean) property, 228
fullscreen config option, 229
fullScreen option, 216
fullScreenClass option, 216

G
geo fix command, 108
GeoLocation class, 110
Geolocation example, Rhodes application framework, 108–111
GET requests, 216
getContacts function, 149
getContacts_callback, 149
glossOnIcon (Boolean) property, 228
Google Maps application, 200
graytitle class, 190
greetingLabel, Interface Builder, 25, 60
greetingLabel IBOutlet, 24
greetingLabel property, 24
greetings text, aligning, 22
Groups & Files section, Xcode, 32–33

H
Hammerschmidt, Christoph, 3
HEAD section, 223
<head> tag, 186, 247
headers, required for iWebKit, 186
Hello BlackBerry Application, 59
Hello iPhone! button, Interface Builder, 25
Hello World application in PhoneGap, 139–140
   Titanium Mobile, 155–157
Hello World project, 54
HelloiPhone file, 32
HelloiPhone project, 19
HelloiPhoneViewController implementation declaration, 24
HelloiPhoneViewController.h file, 23
HelloiPhoneViewController.m file, 24
HelloiPhoneViewController.xib file, 20, 25
Hello.java class, 40–41, 44, 46
helloMessage string, 24
HelloWorld application, 58
HelloWorld class, 55, 57
HelloWorld.java file, 56
HelloWorldScreen class, 57
IBAction keyword, 23–24
IBOutlet keyword, 23
Icon (String) property, 228
icon/ folder, 90
icon option, 216
ID
  finding, 31
  identifying elements in CSS with, 249–250
IDE (integrated development environment), 52
<iframe> element, 241
/images folder, 204
<img> tag, 97, 152
Import 'Button' (android.widget) option, 44
Import wizard, Solution Explorer, 79
index.erb file, 91
index.erb page, 97, 107
index.html file, 138–139, 142–143, 146, 150, 184, 205, 223, 226
index.html web page, 140
index.js file, 226–227, 229
initialization options, for jQTouch, 215–223
initialize method, in RhoSync, 116
initializeTouch option, 216
inline style, 247
inline tags, 239
<input type="checkbox">, 198
<input type="radio"> tag, 198
<input type="tel"> tag, 197
<input type="text"> tag, 197
<input> tag, 199
Install selected button, 109
Installed Components, Eclipse, 55
Installer Package, 134
interactivity, in Sencha Touch, 232
interfaces, creating
  add UI elements, 21
  adding UI elements, 21
  aligning text greetings, 22
  buttons and text field layouts, 22–23
  connecting code to views, 25–26
  overview, 20
  simple user application, 55–57
  Web View, 26–28
  writing controller code, 23–25
internal style sheet, 247
Inventory application, 123
iOS, HTML and CSS support on, 164
iPhone Developer Program Portal, 30
iPhone OS Application option, Xcode, 19, 26
iPhone Simulator menu, 101
iPhone Simulator.app, 208
/iphone/PhoneGap-based Application/www directory, 140
iPhones
  building apps, 18–28. See also interfaces, creating
devvelopment of, 18
installing apps on devices
  creating provisioning profiles, 32
  finding device IDs, 31
  install provisioning profiles, 32
  installing and running on devices, 32–33
  manually setting up provisioning, 30
  using development provisioning assistant, 29–30
PhoneGap, open source framework for, 134–135
  and Rhodes
debugging on, 100–101
running application on, 93
Xcode, 17
iphone/www directory, 142
iPod style, list styles with iWebKit, 193–194
ipodlist class, 188, 194
iScroll library, Cubiq, 180
iTunes, 190–191, 193
iWebKit project, 26
iWebKitDemoViewController.xib file, 26
iWebKit
  body, 186
  forms with, 196–199
  headers required, 186
  and HTML structure, 185–186
  integrating in mobile applications
    adding framework to application
      layout template, 204
    native iPhone application, 201–202
    with Rhodes, 203–204
    setting up PhoneGap for, 205
  and landscape mode, 200
  lists with
    App Store style, 191–192
    classic, 188–190
    iPod style, 193–194
    iTunes classic style, 190–191
    iTunes style, 193
    overview, 187
    navigation with, 194–195
    overview, 183–184
    phone integration in, 200
  iWebKit/app/layout.erb file, 204
iWebKitDemoViewController.h file, 26
iWebKitDemoViewController.m file, 27–28
iWebKitDemoViewController.xib file, 28
adding screens
  with Ajax, 212–214
  buttons on, 215
cancel and back, 214–215
overview, 211
basic views, 217–218
creating simple application, 209–211
customizing application animations, 218
initialization options for, 215–223
integration
  with PhoneGap, 222–223
  with Rhodes, 222
lists, 218–220
overview, 207
running example code, 208
themes in, 221–222
toolbars, 218
jquery.js file, 142–143
JRE (Java Runtime Environment), 52

K
Kaneda, David, 207
Kawamoto, Dawn, 3
Keychain Access, 30
Keytool tool, 50
Kim, Gary, 4
Kitchen Sink application, Titanium, 158

L
Label element, Interface Builder, 22
LabelField class, 57, 60
labels, text fields, and buttons, 58–60
landscape mode, and iWebKit, 200
Landscape view, 233
Launch Assistant, iPhone Developer Program Portal, 30
Launch button, Titanium, 154–155
Layout Editor, 41, 43
Layout menu, Interface Builder, 22
Layout tab, 41
layout.erb file, 91, 204, 242–243
Left Button element, 195
Left Navigation element, 195
leftButton class, 215
<li> tag, 149, 178, 187–188, 199, 218, 220
Library menu item, Interface Builder, 20
Library window, Interface Builder, 20–21, 26
<link> tag, 247
Index

list class, 187
lists
  HTML and CSS support for, 178–180
  with iWebKit
    App Store style, 191–192
    classic, 188–190
    iPod style, 193–194
    iTunes classic style, 190–191
    iTunes style, 193
  overview, 187
  in jQTouch, 218–220
loading.html file, 91
Login screen, 130

M
main method, 57
MainScreen class, 57
main.xml panel, 41
make command, 134
Manage Certificates option, Solution Explorer, 79
Manage tab, iPhone Developer Program Portal, 30
manifest.xml file, 49
map_example/app/Person/person_controller.rb file, 110
MapView class, 110
margin attribute, 251
Market application, 50
Menu button, 42
menus, HTML and CSS support for, 166
metal class, 218
MFC (Microsoft Foundation Classes), 228
Min SDK Version box, 40
mobile applications, 6–10
models, generating in Rhodes, 95–98
Model-View-Controller (MVC), 18, 83–84, 95
musiclist class, 187
MVC (Model-View-Controller), 18, 83–84, 95
MyCompany, Registry Editor, 78

N
Name field, Titanium, 154
Name value, Registry Editor, 78
nameFilter parameter, 149
Native Development Kit (NDK), 36, 39
native picker control, iOS, 171
native UI elements, and runtime architecture, 86
navigation
  bars, HTML and CSS support for, 168–180
  with iWebKit, 194–195
navigator.camera.getPicture function, 152
navigator.contacts.displayContact function, 149
navigator.contacts.getAllContacts function, 149
navigator.contacts.newContact function, 149
NDK (Native Development Kit), 36, 39
New App ID button, iPhone Developer Program Portal, 31
New email application, 200
New Key option, Registry Editor, 78
new person form, 109
New Project dialog box, Eclipse, 53
New Project icon, Titanium, 154
New Project menu item, Visual Studio 2008, 77
New Project window, Visual Studio 2008, 67
New Referencing Outlet, Interface Builder, 26
new.erb page, 110
Nielsen, Jacob, 6
NSURLRequest, Xcode, 28
numberfield xtype, 230

O
OAV (object-attribute-value), 114
Object Relational Manager (ORM), 84
object-attribute-value (OAV), 114
OHA (Open Handset Alliance), 35
onClick callbacks, 149
onClick event, 45, 160
onClick method, 45
onCreate method, 40, 44–45
onKeyUp event, 150
onReady (Function) property, 228
onReady property, 227
Open Handset Alliance (OHA), 35
Options menu, 130
Organizer window, Xcode, 31, 33
ORM (Object Relational Manager), 84
OTA (over the air) distribution, 63
Other Project Types, Visual Studio 2008, 77
Outline tab, 43
Output file name field, Solution Explorer, 77
over the air (OTA) distribution, 63

**P**

*p* elements, 248
*<p>* tag, 248, 252
Package menu item, Eclipse, 55
padding attribute, 251
page-home div, 212
pageitem class, 190, 199
pages, HTML, 75
pageSize option, 149
/palm/framework/www directory, 140
panel xtype, 230–231
pause/play icon, 194
Perez, Bryan, 6
Personal Information Management (PIM), 103, 146
PersonController class, 110
pg_camera project, 150
pg_contacts project, 146, 152
Phone application, 200
PhoneGap
  camera example, 150–152
  contacts example, 146–150
  Hello World application in, 139–140
  overview, 131–133
  PhoneGap simulator for, 138
  setting up for
    Android, 136
    BlackBerry, 137
    iPhone, 134–135
    iWebKit, 205
tip calculator example, 141–145
PhoneGap Simulator, 138, 146
PhoneGap.addConstructor function, 149
phonegap/android directory, 136
PhoneGap-based Application option, Xcode, 134
phonegap/blackberry/framework/src/www/directory, 144
phonegap/iphone folder, 134
phonegap.json file, 137
phonegap.js file, 149
PhoneGapLib library, 134
PhoneGapLibInstaller.pkg file, 134
phoneStartupScreen (String) property, 228
photo_controller.rb file, 107
PIM (Personal Information Management), 103, 146

**P**

Placeholders attribute
  HTML5, 173
  Interface Builder, 23
plastic class, 218
platform, 51–52
popSelector option, 216
Portrait view, 233
preloadImages (Array) property, 228
preloadImages option, 216
Pretty format, 188
Product edit page, 100
product inventory example, in RhoSync
debugging source adapters, 130
implementing source adapter, 126
overview, 122–125
setting up RhoSync server, 129
testing application, 130
testing source adapter, 126–130
product names, customizing, 77
product_controller.rb file, 96
product_spec.rb file, 128
ProductName field, Visual Studio 2008, 77
product.rb file, 96, 126, 128
product.rb source adapter, 126
Products link, 97
Program Portal, 31
Programs Folder, File System Editor, 78
Progressive Disclosure, 166
Project creation dialog box, Eclipse, 54
Project drop-down list, File System Editor, 78
Project Type field, Titanium, 154
Project Types pane, Visual Studio 2008, 67, 77
projects, Eclipse, 53–55
Properties context menu item, Solution Explorer, 77–78
Properties pane, Visual Studio 2008, 70, 73
Properties panel, 43
Properties section, solution browser, 75
Properties tab, 43, 76
Properties Window
  Registry Editor, 78
  Visual Studio 2008, 77
provisioning
  manually setting up, 30
  profiles, creating, 32
public/ folder, 90
public void fieldChanged(Field field, int context) method, 60
Publisher URL field, Titanium, 154
pushScreen() method, 57
puts @result.inspect statement, 130

Q
query method, in RhoSync, 117–118

R
radio buttons
  HTML and CSS support for, 175–177
select modal view, Android, 171
Radio xtype, 231
radiobutton class, 198
Rails. See Ruby on Rails, and Rhodes
rake clean:android command, 92
rake clean:bb command, 92
rake clean:iphone command, 92
rake clean:win32 command, 92
rake clean:wm command, 92
rake commands, 92
rake device:android:debug command, 92
rake device:android:production command, 92
rake device:bb:debug command, 92
rake device:bb:production command, 92
rake device:iphone:production command, 92
rake device:wm:production command, 92
rake program, 91
rake run wm:dev command, 92
rake run wm:devcab command, 92
rake run wm:emu command, 92
rake run wm:emucab command, 92
rake run:android command, 92
rake run:android:device command, 92
rake run:bb command, 92, 95
rake run:iphone command, 92–93
rake run:wm:emu command, 95
rake task, 91
rake uninstall:android command, 92
rake uninstall:android:device command, 92
Rakefile file, 90
rake:run command, 98
Received Actions, Interface Builder, 25
Registry entries, adding, 78
Remove context menu item, 42

Request Certificate button, iPhone
  Developer Program Portal, 30
Research in Motion (RIM), 51, 165, 235
Reset Content and Settings... menu item,
  iPhone Simulator menu, 101
resignFirstResponder method, 25
res/layout/main.xml file, 40, 42
resources directory, 186
rhoconfig.txt file, 90, 100, 103, 107
Rhodes application framework
  building application, 89–91
  Camera example, 106–108
  Contacts example, 103–106
  database for (Rhom), 86–87
document in
  on Android, 101
  on BlackBerry, 101
  on iPhone, 100–101
development architecture for, 84–85
device capabilities with, 101–102
dynamic layout with, for BlackBerry
  browser-rendering engine 4.2, 242–243
documentation
example applications
  Camera, 106–108
  Contacts, 103–106
  Geolocation, 108–111
generating model in, 95–98
Geolocation example, 108–111
installing, 88–89
integrating iWebKit in mobile
  applications with, 203–204
overview, 83–84
and Ruby on Rails, 88
running application
  on Android, 94
  on BlackBerry, 94–95
  on iPhone, 93
overview, 91–92
  on Windows Mobile 6, 95
runtime architecture for, 85–86
threading in, 87
rhogen app command, 89
rhogen model command, 95, 106
RhoLog.txt file, 100
Rhom, database for Rhodes, 86–87
Rhomobile app directory structure, 88
Rho::RhoContact.find(:all) function, 104–105
RhoSync
  authenticate method, 121
  authentication in, 116–117
create method, 119
data storage format in, 114
delete method, 120
initialize method, 116
methods in
  authenticate, 121
create, 119
delete, 120
initialize, 116
query, 117–118
sync, 119
update, 120
overview, 113
product inventory example
debugging source adapters, 130
generating RhoSync application, 128
implementing source adapter, 126
overview, 122–125
setting up RhoSync server, 129
testing application, 130
testing source adapter, 126–130
query method, 117–118
source adapters in, 115
sync method, 119
update method, 120
rhosync/lib directory, 128
rhosync/vendor/sync directory, 128
RichTextField class, 57, 61
Right Button element, 195
Right Navigation element, 195
RIM (Research in Motion), 51, 165, 235
R.layout.main parameter, 40
rounded class, 219
Ruby on Rails, and Rhodes, 88
Run menu, 41
Run on Device screen, Titanium, 157
runtime architecture, for Rhodes, 85–86

S

SaaS (Software As A Service), 113
SampleWebView.java class, 48
sayHelloToUser method, 24–25
scaffold-generated app, 24–25
Scalable view, 166
schema.rb file, 88
scope (Object) property, 228
screens
  adding with jQTouch
    with Ajax, 212–214
    buttons on, 215
cancel and back, 214–215
  resolutions, for BlackBerry, 244–245
Scroll view, 166
SDK directory, 37
SDK Setup.exe file, 37
SDK tools/ directory, 49
Search box, Interface Builder, 21
searchbox class, 186
Select Certificate window, Solution Explorer, 79
Select class, 198
select xtype, 230
<select> tag, 198
selection boxes, HTML and CSS support for, 171–173
Sencha Touch
  adding components in, 231
  interactivity in, 232
  overview of framework, 227–231
  setting up, 225–227
Sencha Touch library files, 227
Server tab, RhoHub, 126
servers, RhoSync, 129
setChangeListener(this) method, 60
setContentView method, 40
Settings, RhoHub, 125
Settings drop-down menu, Interface Builder, 20
/Settings folder, 91
Settings screen, 90
Setup and Deployment, Visual Studio 2008, 77
shortcuts, applications, 78
Show Develop menu in menu bar option, 212, 229
Show Error Console menu option, 229
Show Records option, RhoHub, 126
Show Web Inspector option, 229
showContact function, 149
Simple format, 188
Simple Tip Calculator Application for BlackBerry, 144
Simulator option, 135
slider xtype, 230
slideSelector option, 217
slideupSelector option, 217
SMALL element, 220
smallfield class, 197
Smart Device Cab project, Solution Explorer, 78
Smart Device CAB Project template, Visual Studio 2008, 77
Smart Device option, Visual Studio 2008, 67
Smart Device Projects, creating, 67
SmartDeviceProject1 project, 72
smartphones
  application marketplace, 2–4
cross-platform frameworks, 5–6, 10–13
mobile applications, 6–10
overview, 1
web techniques, 10
SMS application, 200
SOFTWARE, Registry Editor, 78
Software As A Service (SaaS), 113
Solution Explorer, 78
source adapters, in RhoSync
debugging, 130
implementing, 126
overview, 115
testing, 126–130
source class, 114
SourceAdapter class, 114
sources subdirectory, 128
spacer xtype, 230–231
<span class="graytitle"> tag, 199
<span class="number"> tag, 194
<span> tag, 190, 194
spec/sources/ subdirectory, 128
splitbutton component, 231, 233–234
src directory, 40, 44
Start Menu Folder, File System Editor, 78
startupScreen option, 217
stash_result function, 119
statusBar option, 217
statusBarStyle (String) property, 228
<style> tag, 247
Submit button, Visual Studio 2008, 74
submitButton handler, Visual Studio 2008, 74
submitForm function, 149
submitSelector option, 217
super() method, 57
swapSelector option, 217
/symbian.wrt/framework/www directory, 140
class.sync method, in RhoSync, 119
syntax, for CSS, 248–249

tabBarHidden property, Titanium, 160
tabpanel xtype, 230
tail -f command, 101
takePicture function, 152
target attribute, 214
Target Machine pane, File System Editor, 78
Targets drop-down, Xcode, 33
Task details page, 99
Tasks list page, 98
Tasks new page, 99
Templates pane, Visual Studio 2008, 67, 77
Terminal.app application, 134
Test & Package tab, Titanium, 154–155
text areas, HTML and CSS support for, 174–175
text boxes, HTML and CSS support for, 173–174
text fields, 22–23, 58–60
text greetings, aligning, 22
Text property, 44
text-align attribute, 252
textarea xtype, 231
Textbox class, 199
text-decoration attribute, 252
textfield xtype, 231
text-transform attribute, 252
TextView control, 44–45
theme.css file, 221
themes, in jQTouch, 221–222
themes/ directory, 222
themes/apple/theme.min.css file, 211
themes/jqt/theme.min.css file, 211
threading in Rhodes, 87
tip calculator example, in PhoneGap, 141–145
Titanium Mobile
  building applications in, 157
  Camera example, 158–160
device capabilities in, 157–158
  Hello World application, 155–157
  overview, 153–154
Titanium.UI module, 157
Titanium.UI.AlertDialog class, 157
Titanium.UI.Android module, 157
Titanium.UI.Button class, 157
Titanium.UI.iPhone module, 157
Title element, 195
to_s method, 110, 118
toolbar component, 234
toolbar xtype, 230
toolbars
  HTML and CSS support for, 167
  in jQTouch, 218
toolbox, Visual Studio 2008, 73
Toolbox pane, Visual Studio 2008, 69, 73
Tools menu
  Interface Builder, 20, 22, 25–26
  Visual Studio 2008, 76
topbar class, 186, 194
Touch Up Inside option, Interface Builder, 25

U
UDIDs (Unique Device Identifiers), 30
UI elements
  adding, 21
  native, and runtime architecture, 86
UI widgets, HTML and CSS support for, 169
UiApplication class, 57, 59
UIWebView, Interface Builder, 27
<ul class="pageitem"> tag, 190
<ul> tag, 178, 187–188, 190, 199, 218
Unique Device Identifiers (UDIDs), 30
update method, in RhoSync, 120
update_hash parameter, 120
update(txt) function, 232
USB Debugging check box, 49
Use Current Location check box, 109–110
useAnimations option, 217
user interaction, for BlackBerry, 244–245
User Templates section, 134
UserInterface class, 59
UserInterfaceScreen class, 60
userNameField class, 24–25, 60
Users tab, RhoHub, 125

W
W3C (World Wide Web Consortium), 131
Wasserman, Todd, 6
web techniques, 10
Web View
  HTML and CSS support for, 178–182
  Xcode, 26–28, 75
_webapp target, 218
WebBrowser controls
  adding, 75
  loading HTML in, 76
WebBrowser element, Visual Studio 2008, 76
WebKit web views, HTML and CSS support for, 178–182
-webkit-appearance property, 173
WebView, embedding in application, 46–48
WebKit.navigate method, 108
Widarsson, Fredrik, 3
width property, 44
Windows Marketplace for Mobile, 76, 80
Windows Mobile
  building apps. See also base functionality
    adding WebBrowser controls, 75
    creating HTML pages, 75
    creating Smart Device Projects, 67
    deploying and testing, 72
    embedding Web View in applications, 75
    loading HTML in WebBrowser controls, 76
distributing applications, 80
HTML and CSS support on, 165
overview, 65
packaging and distributing apps
  adding applications to CAB Projects, 78
  adding CAB Projects to solutions, 77
  adding Registry entries, 78
building and deploying CAB files, 78–79
creating application shortcuts, 78
customizing product names, 77
installing CAB files, 79
overview, 76
version 6, 95
version 6.5 development, 66
Windows Phone Marketplace, 66
/winmo/www directory, 140
withimage class, 188
Wolfe, Alexander, 4
World Wide Web Consortium (W3C), 131
WWDR (Worldwide Developers Relation), 30

www directory, 139, 146, 150, 205, 222
www/index.html file, 140

X
Xcode, 17–19
Xcode file copy prompt, 202
Xcode groups option, 200–201
.xcodeproj file, 157