Mobile Websites With Minimum Effort

That the future is mobile is not news. That libraries need to have a presence in the emerging mobile landscape is also not news. The question is how do you move there? It's not as if your regular library website is going to start testing, updating, and redesigning itself: beleaguered webmasters need to find a way to develop a strong mobile presence with minimal effort.

PARTY LIKE IT’S 1999

What's old is new again—and that's not great news. The desktop world in the last few years evidenced a gradual but commendable move toward more standards compliance among browsers rather than less. In the mobile browser world, it's 1999 all over again. Indeed, usability expert Jakob Nielsen recently described mobile usability as “miserable” and likened it to the state of usability on the desktop a decade ago (www.useit.com/alertbox/mobile-usability.html). That means varying degrees of standards compliance, cross-platform inconsistencies and incompatibilities, and some good old-fashioned, outright deception on the part of mobile browsers.

Are we back to the days of coding multiple versions of our pages for various browsers—tweaking, hacking, and praying along the way? Thankfully no, but webmasters do need to make some choices.

PICK YOUR CONTENT

Mobile users face a number of constraints in getting to content while on the go. First, a fast mobile connection is slow by desktop standards. Second, even on relatively spacious handhelds such as the iPhone, screens are small. Every pixel counts. Navigation and input take effort, even on the largest of touchscreen displays such as that of the BlackBerry Storm, the myTouch, or the iPhone. Last but by no means least, mobile users are, by definition, out of their offices and libraries; away from their desks. As the W3C says in the Mobile Web Best Practices document (www.w3.org/TR/mobile-bp), “Mobile users ... are likely to have more immediate and goal-directed intentions than desktop Web users. Their intentions are often to find out specific pieces of information that are relevant to their context.”

To put it another way, no mobile user wants to read your circulation policy on his or her handheld while standing in line at the dry cleaners, but he or she does want to know if you’re open later than 5 p.m.

Focus, then, on a mobile site that provides information and services that might be useful to a person in motion: directions, library hours, contact information, a
simple catalog search, item holds, and renewals. The “Ask a librarian” and “Text a librarian” features are also obvious candidates for inclusion. It is not worth the effort to make your entire site mobile; that’s counterproductive. Go for simple and straightforward.

Since you’ll be serving only select portions of your site for mobile consumption, it makes sense to create a new mobile-optimized homepage. It’s a pretty straightforward affair and will most likely be faster than trying to retrofit your existing homepage for mobile. Below the level of the homepage, however, it’s probably best to serve existing content from your website that is reformatted and optimized for mobile. I don’t think any webmaster has an interest in trying to maintain two versions of the library’s hours of operations page, contact page, or pretty much any other page for that matter.

OPTIMIZING THE EXPERIENCE

Smartphones are the future and, at least in North America, represent the largest area of mobile growth. Thus, it makes sense to design with an eye toward optimizing the experience for the smartphone user. What of non-smartphone users? Users with smaller screens and slower connections will still get a usable experience. But designed correctly, one site can serve a usable experience to all users and still deliver an optimized experience to smartphone users.

Thankfully or not, depending on your perspective, the WebKit-based Safari browser for the iPhone alone represents more than 50% of North American mobile internet traffic. The default browser for Android-based devices such as the Google phones also happens to be WebKit-based. In August 2009, BlackBerry maker RIM acquired a mobile web browser company, Torch Mobile, that bases its product on WebKit; the Symbian platform used on Nokia devices also has a WebKit-based browser.

What does this mean? While each individual implementation of WebKit varies a great deal, there are commonalities. Those four smartphone providers represent much more than 90% of the North American smartphone industry. So designing something that looks and works well across WebKit is designing with most of the smartphone market in mind.

If you’re designing a new mobile-optimized homepage from scratch, then by default, and by using some of the coding tips listed here, you’ll be good to go on mobile devices. What about existing pages on your site that you’d like to serve mobile users? The CSS-savvy among you might be wondering, “Why not just use CSS media types to identify a style sheet for mobile, a different one for a desktop browser, and perhaps another for print? It’s been around for years and years, right?” This is where the deception comes into play. In order to give their users the “full web experience” that device makers tout, some mobile browsers don’t identify themselves as “mobile” and ignore mobile style sheets. Sneaky! What to do?

UNDER THE HOOD

There are two CSS techniques that are incredibly useful tools to have in your “going mobile” toolkit. The first is something called CSS media queries. As I said earlier, it’s unfortunately not enough to have a mobile style sheet available and to direct mobile browsers to it by using the media type “handheld” since many mobile browsers simply ignore that declaration. While media queries are not part of the official CSS3 specification yet, it is a W3C “candidate
recommendation,” a document that has been widely reviewed and ready for implementation, and so is standards-compliant (www.w3.org/TR/css3-mediaqueries).

A media query looks like this:

```html
<link media="only screen and (max-device-width: 480px)" href="/mobile/mobile.css" rel="stylesheet" type="text/css"/>
```

This query says, “I know that even though you're a handheld, you don't call yourself a handheld. However, if the screen on which this page is going to be displayed is 480 pixels wide or less, use the mobile style sheet.” You don't need the CSS declaration on your new mobile homepage; you've created that from scratch, so it's clean, small, simple, and text-based already. However, this media query is very useful for pages appearing on your existing site that you also want to serve to mobile—an example might be your library hours page. Let's assume that page has navigation tabs, a header, a footer, and maybe an image or two. You don't want to create a new mobile-optimized version of this page from scratch, and you certainly don't want two versions of this page floating around. Insert the media query statement, and voilà! The page now displays fully on any screen that's wider than 480 pixels. On handhelds it uses your mobile style sheet to display properly. So now that you've gotten the mobile browser to point to your mobile style sheet, what should it contain?

I'm making an assumption here that you're already using CSS for both the style and the layout of your existing pages. If you're not, well, that's another article entirely. Your best friend here is the CSS statement `display:none`. You can apply this statement to any HTML DIV tag to keep the stuff within that DIV (section/division) from displaying. So those navigation tabs, headers, footers, call-out boxes, and the like can be turned off in your mobile style sheet:

```
#header {display: none;}
```

**MOBILE SITE CODE TIPS AND TRICKS**

- Using well-formed code is important for your standard site; for your mobile site it's critical. Validate and test your mobile pages using the W3C's MobileOK Checker (http://validator.w3.org/mobile).

- Be sure to use the mobile document type declaration (DTD) for any mobile-only page, such as your mobile homepage:

  ```xml
  <!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN" "www.wapforum.org/DTD/xhtml-mobile10.dtd">
  ```

- Use the HTML `<accesskey>` command for elements that allow for user input such as links, buttons, and text input boxes. When mobile users scroll through a page, these access-keyed items will gain focus, making navigation easier. Number them sequentially as they appear on the page.

```html
<a accesskey="1" href="www.library.pitt.edu/libraries/hours/all.html">Hours</a>
```

- Most smartphone mobile browsers parse pages looking for what they think are telephone numbers and make them actionable, meaning they can be clicked to initiate a phone call. You can help this process along and increase usability by using an a href="tel:" statement like this:

  ```html
  <a accesskey="3" href="tel:(123) 555-1212">(123) 555-1212</a>
  ```

where you include your phone number.

- On the desktop, it's considered best practice to have consistent, persistent navigation options on every page; on mobile pages that usually takes up too much real estate. One link back to home on every page should suffice.

- If you ask for feedback, be sure to grab users' user agent strings. If they report a problem with your mobile site, this is very useful information to have. You can grab the user agent using a lightweight PHP or JavaScript statement.

  ```javascript
  var userAgent = navigator.userAgent;
  ```

- Put your content in a single, left-aligned column.

- Mobile devices are often used outdoors. Glare can wash out the screen, so think high-contrast for your color scheme.

- Be careful about handoffs to nonmobile optimized pages. If it's not a mobile-friendly site, think twice about sending your users there.

The last tip leads to the question of the library catalog, an obvious candidate for inclusion in your mobile site. The short answer is most catalogs are not very mobile-friendly. One route is to develop a library catalog app, as opposed to simply linking to your catalog. But, of course, that requires that you or your shop has the expertise and resources to create an app. Ideally, vendors will create mobile-friendly versions of their products. One or two vendors have done this, but it's not yet commonplace. A good interim solution might be to see if there's an accessible version of your catalog. These text-based interfaces are now common, and an out-of-the-box version will likely be more mobile-friendly than your standard OPAC interface. If you can configure it, keep this list of tips and tricks in mind to optimize the user experience.
USABILITY TESTING

Before you launch, don’t forget to test your mobile site on as many platforms as you can. There’s a good chance that you’ll have a good representation of mobile platforms and browsers among your co-workers. So be sure to put the site through its paces.

A few thoughts on mobile usability field testing: Unlike our standard sites, our mobile sites aren’t, or should not be, very complex. Whereas in the desktop world there may be many different paths (and, consequently, many rabbit holes down which a user can fall), our mobile sites should by design be streamlined. In addition, unlike desktop usability studies where the platform on which the testing is done is uniform, that’s not going to be the case for mobile devices, unless your library has a stash of out-of-the-box iPhones, myTouches, and Storms lying around. On the desktop we have usability software to help us capture and analyze usability test participant behavior but no equivalent for all the smartphone platforms as of yet.

Should you not bother then to test your site with users? I’d say, yes, do test. Any testing you do will probably be less formal, and less rigorous, than what we’re able to do on the desktop—but we should do it nonetheless.

TOOLS

To learn more, take a look at the mobile websites of some other libraries:

• New York Public Library Mobile (http://m.nypl.org)
• University of Virginia Library Mobile (http://uvalidmobile.appspot.com)
• Boston University Medical Center Mobile Library (http://medlib.bu.edu/mobile)

Here are some other sites and tools of interest:

• Mobile Web Best Practices W3C (www.w3.org/TR/mobile-bp)
• Mobile Web Test Suites Working Group (www.w3.org/2005/MWI/Tests)
• MobileOK Checker—A mobile webpage validator for the W3C (http://validator.w3.org/mobile)
• BrowserCam Device Capture (www.browsercam.com/De fault2.aspx) (In addition to desktop browsers, BrowserCam has emulators for mobile browsers.)
• Google Webmaster Tools: Developing Mobile Sites (www.google.com/support/webmasters/bin/answer.py?answer=72462)

The library webmaster these days wears many, many hats. We need to be mobile, and we need to do it sooner rather than later, but with a little creative re-purposing, reusing, and recoding, we can be there without breaking the bank.

Jeff Wisniewski (jeffw@pitt.edu) is web services librarian, University Library System, University of Pittsburgh.
Comments? Email the editor (matydee@xmission.com).