# Table of Contents

Introduction ........................................................................................................... 3  
Think lean, think streamlined ............................................................................. 4  
Maintaining the identity of your brand ............................................................ 6  
Minding limitations ......................................................................................... 7  
Layout ................................................................................................................ 8  
Page consistency ............................................................................................. 8  
Headers and footers ....................................................................................... 9  
Stylesheets ..................................................................................................... 10  
Grouping phones .......................................................................................... 10  
  Screen Classifications and Recommended Design Specifications .............. 11  
  WAP 2.0 Phone Models and their Screen Classifications ......................... 12  
Designing for multiple phone groups ............................................................ 13  
Graphics Guidelines ..................................................................................... 14  
  Tips for compressing images .................................................................. 14  
  Tips for presenting images .................................................................... 14  
  Heavy image use .................................................................................... 15  
Navigation ..................................................................................................... 16  
  You are wedded to the 5-way .............................................................. 16  
  Pagination of extended content .......................................................... 18  
  User input ............................................................................................... 18  
  Access keys ............................................................................................ 19  
  Miscellaneous navigational tips ............................................................ 20  
Conclusion .................................................................................................... 21  
Appendix: Code samples ............................................................................. 22  
  Embedded telephone number ............................................................... 22  
  -wap-input-format ............................................................................... 23  
  Prefetch ................................................................................................. 24  
  Transparent PNGs ............................................................................... 25
Introduction

With the transition from WAP 1.0, with its monochromatic text and limited image support, to WAP 2.0, mobile web developers have seen their arsenal expand dramatically. WAP 2.0, with its support for stylesheets, XHTML formatting elements, and graphics, is a major step forward for standardized markup browsing on handheld devices. While these new capabilities allow the designer to approach the desktop web experience, they do not allow one to fully emulate it, and nor should it be aspired to – browsing from a mobile phone brings with it not just a host of device constraints and hardware variability, but also an entirely different set of use cases, user demands, and opportunities. To that end, this style guide will not just attempt to advise on development for the mobile browser, but also the mobile user.

This document is intended to be a basic, high-level conceptual guide – an introduction. It will refer to other documents included with your content provider WAP package for more technical detail.
Think lean, think streamlined

To start with, here’s just a short example list of things you won’t be able to or shouldn’t do with a mobile WAP 2.0 browser:

- Client-side scripting – no Javascript or any other type of client-side scripting to validate forms, change images, etc.
- Flash – no animations.
- Multi-column layouts – limited screen size will prevent you from including sidebars or other navigation blocks alongside your main content. Most phones will use a narrower portrait format instead of the typical PC landscape.
- Complex forms – while the browsers will support as complex and lengthy a form as you can compose, users won’t be inclined to enter large amounts of data from their handsets. Asking for an email address is fine. Asking for an email address, postal address, first name, last name, blood type, eye color, marital status, top ten favorite albums, and life history is pushing it. Not only is text entry more difficult, but consumers may feel that information entered on the phone is more personal.
- Incorporate large amounts of information on a single page - even the largest phone displays will have but a fraction of the typical desktop screen space.

An immediate reaction may be something along the lines of “My goodness! There is simply no way I can create an effective representation of my company or product with such harsh restrictions.”

Don’t think of these as limitations – think of them as thankfully absent distractions. By focusing and honing your intended content you’ll be able to deliver a much more usable browser application. Remember that what you trade off in sophistication you more than make up for in the fact that your site will be in your customers’ hands at all times.

Let’s take a look at the example web site on the following page:
The original page contains far too much information to be presented on a small screen as it is, but by extracting some core elements and working within the narrow, vertically-oriented structure of the WAP browser, we can create a very functional equivalent.

The Customer Results logo has been resized and combined with brand-consonant imagery (shopping carts) at the top of the page. Each of the banner elements (categories 1 through 9) has been combined together into a single “Categories” item. The subsequent headers have all been given their own link, followed by a promotional image. Lastly, two footers, one for the site and the other for VZW Today, have been placed at the bottom.

Before delving into some details about how your WAP site might be designed, let’s review some general principles to be followed throughout the entire process.

**Maintaining the identity of your brand**

Many of the tools a desktop web designer might typically use to establish a memorable and cohesive brand identity will be unavailable for WAP. Thus simpler techniques become all the more important. As most browsers will come equipped with only a single font, paying attention to text color, background color, and placement is vital. As a result of limited screen resolution and memory size, a logo or icon with few colors will prove far more effective than a complicated photo-realistic image occupying a majority of the screen. You may find that, finally free from chaotic clutter, consistently sticking with a few judicious branding elements will actually solidify and enhance the message you’re trying to articulate. The limited form factor will force you back to basics, but very simple touches will yield effective results. Think of it as a combination of boot camp and haiku.

Because the “little things” constitute the entirety of what you can do to maintain your brand identity, there’s no excuse to forget or ignore them.
Minding limitations

While WAP is an amazing platform with which one can deliver a variety of content, build applications, and put a site in front of more customers than ever before, it would be downright disingenuous not to acknowledge that it has limitations in spades. As stated before, if respected they can only help focus presentation and development, but if ignored they can quickly torpedo the best of concepts.

These shortcomings may be intuitive, but each has some corollaries that deserve elucidation.

- Limited size – even at QVGA resolution (the highest handset resolution available today – equal to ¼ of a VGA desktop screen, or 240 x 320 pixels), your space will be restricted. Obvious effects of this include being unable to present large images or detailed navigational elements. Less obvious are the expectations of your users. Links below the fold may go unseen. Articles requiring excessive scrolling may be abandoned. Worse yet, articles requiring a user to page through lengthy headers may go unread entirely.
- Limited bandwidth – while EVDO handsets are becoming more prevalent, the bulk of your customer base will be using devices that load pages at speeds comparable to dial-up. Furthermore, they will have increased latency, meaning that not only does the overall size of the page matter, but so does the number of its elements; a page with five 1k images will load much less quickly than a page with one 5k image. When possible (the viewing of a multi-page article, for example) use pre-fetching to cut down on delays. If more than a few images must be loaded, use the multipart MIME transfer method outlined in the separate PDF (“WAP 2 MIME Multipart”).
- Limited memory – while the dangers of graphics-intensive pages have already been mentioned in the above bullet point, the limited memory of most devices additionally results in little or no caching by the browser. Do not assume that when quickly navigating back and forth between a couple pages the browser will be unburdened by network connections.
- Limited navigation – with only a five-way keypad to highlight links, a twelve-button numerical pad to input text, and limited use of menus, sites must keep in mind the inability of the use to browse as fluidly as on the desktop.
- Limited session time – complicated transactions and lengthy interaction are less likely to be undertaken by users on the go. Part of making your site streamlined for WAP includes making useful functionality quickly accessible.
Most of these limitations crop up only when testing on actual phones as opposed to emulators. Heavy testing on real handsets up-front will keep these requirements in the forefront of your mind.

Layout

Page consistency

Thankfully, this is one consideration that doesn’t differ much from desktop web design, or even general application design. The elements at your disposal need to be used in a consistent and distinguishable fashion. In our example site, we’re presenting a home page, category list page, article page, and image page. Let’s take a look at how they’re tied together visually.
Every page contains a header graphic, content, and two footers. Article and photo pages contain navigational links after the main content to help direct users back to whence they came or onward to additional information. Body links are consistently colored, and home/category navigational links are clearly laid out in a single-column list.

Note that in our example the photo page has been separated from the article content.

Much like a desktop browser, keeping central navigation and/or site features “above the fold” is an important consideration, but with much less screen real estate the task presents different challenges on a mobile device. Be aware of page breaks and how they vary between handsets. In the example above, key information has been placed above the 146 pixel page-break mark. While the home page banner serves as a welcome image, and is thus larger, subsequent banners occupy less than 10% of the screen real estate available above the fold.

Headers and footers

One key aspect of navigational and visual consistency is the use of one or more headers and/or footers. Unlike the expansive headers, side bars, and footers used for desktop web sites, these need to be concise. For example, our “Customer Reports” site uses only a small logo graphic as a header, and a simple, visually-distinct home link for a footer. While such links can technically go anywhere on the page, the scarcity of screen space above the fold makes the footer a superior place for their placement.
In our example Customer Results site, we’ve used two links. The first is a link to the WAP application’s home screen, and the second is a link to a separate site-specific help page.

A Verizon Wireless footer should be placed as the last element on every page, with background color #D80000 and text color #FFFFFF. The footer should contain a single link, named “VZWToday Home”, and linking to: http://wap.vzw.mcore.com.

Stylesheets

Just as consistent desktop web design relies heavily on stylesheets, so should WAP site design. WAP 2.0 uses a subset of CSS2, and adds some phone-specific properties as well. The complete specification can be found at:


Note that some commonly relied-upon components of CSS for the desktop are unavailable. The :hover pseudo-class, for example, most often used to style links over which the cursor is placed, is not mandatory in the specification and remains unsupported in the Openwave browser installed on most Verizon Wireless handsets.

Grouping phones

Fixed-point font sizes and any graphics you decide to incorporate will not translate well between handsets. A 112-pixel-wide header graphic may occupy the entire width of an LG VX4500, but it will cover less than half the width of a QVGA handset like the Audiovox 9900. To combat this problem, we recommend categorizing devices by screen size according to the table on the next page. This provides the optimal balance of minimizing redundant work and ensuring that your site will appear tailored to each phone.

You will need to recognize the phone by its User Agent (UA) string to dynamically serve the appropriate page elements. Be advised: searching for a complete UA string may under-match, as some include the version number of the browser or software build. For instance, matching the entire string

LGE-VX7000/1.0 UP.Browser/6.2.3.1.174 (GUI) MMP/2.0

Would fail to catch

LGE-VX7000/1.0 UP.Browser/6.2.3.2.174 (GUI) MMP/2.0

Please refer to the “WAP 2.0 Phones-UA string_Display Dimension” spreadsheet enclosed with your Verizon Wireless WAP developer package for a list of current user agents.
Screen Classifications and Recommended Design Specifications

<table>
<thead>
<tr>
<th>Class</th>
<th>Total Display</th>
<th></th>
<th>Usable Area</th>
<th></th>
<th>Recommended Design to</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wt</td>
<td>Ht</td>
<td>Wu</td>
<td>Hu</td>
<td>Wt</td>
</tr>
<tr>
<td>Small</td>
<td>120 to 160</td>
<td>128 to 160</td>
<td>112 to 151</td>
<td>89 to 146</td>
<td>112</td>
</tr>
<tr>
<td>QCIF(^{1})</td>
<td>176</td>
<td>220</td>
<td>167 to 169</td>
<td>183 to 186</td>
<td>167</td>
</tr>
<tr>
<td>QVGA(^{2})</td>
<td>240</td>
<td>320</td>
<td>230 to 240</td>
<td>234</td>
<td>230</td>
</tr>
<tr>
<td>QVGAL(^{3})</td>
<td>320</td>
<td>240/256</td>
<td>310</td>
<td>208 to 220</td>
<td>310</td>
</tr>
</tbody>
</table>

All dimensions in pixels

1) QCIF = Quarter Common Intermediate Format = a common standard size for video cameras 176 x 144 pixels
2) QVGA = Quarter VGA = one quarter of the VGA PC monitor standard, in portrait format: 240w x 320h.
3) QVGAL = Quarter VGA Landscape – same size as QVGA only presented in landscape format: 320w x 240h.
## WAP 2.0 Phone Models and their Screen Classifications

<table>
<thead>
<tr>
<th>Manu/Model</th>
<th>Launch Date</th>
<th>Class</th>
<th>Total Display Width $W_t$</th>
<th>Total Display Height $H_t$</th>
<th>Vertical Scroll Bar? Y/N</th>
<th>Usable Width $W_u$</th>
<th>Usable Area Height $H_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG VX4500</td>
<td>Jan-04</td>
<td>Small</td>
<td>120</td>
<td>160</td>
<td>Yes</td>
<td>8</td>
<td>112</td>
</tr>
<tr>
<td>Samsung A790</td>
<td>May-04</td>
<td>QCIF</td>
<td>176</td>
<td>220</td>
<td>Yes</td>
<td>7</td>
<td>169</td>
</tr>
<tr>
<td>LG VX7000</td>
<td>Jul-04</td>
<td>QCIF</td>
<td>176</td>
<td>220</td>
<td>Yes</td>
<td>7</td>
<td>169</td>
</tr>
<tr>
<td>Motorola 810</td>
<td>Aug-04</td>
<td>Small</td>
<td>128</td>
<td>160</td>
<td>Yes</td>
<td>9</td>
<td>119</td>
</tr>
<tr>
<td>Motorola V710</td>
<td>Aug-04</td>
<td>QCIF</td>
<td>176</td>
<td>220</td>
<td>Yes</td>
<td>8</td>
<td>167</td>
</tr>
<tr>
<td>Audiovox 8910</td>
<td>Oct-04</td>
<td>Small</td>
<td>128</td>
<td>160</td>
<td>Yes</td>
<td>9</td>
<td>119</td>
</tr>
<tr>
<td>Audiovox 9900</td>
<td>Oct-04</td>
<td>QVGA</td>
<td>240</td>
<td>320</td>
<td>No</td>
<td>NA</td>
<td>240</td>
</tr>
<tr>
<td>LG VX6100</td>
<td>Oct-04</td>
<td>Small</td>
<td>128</td>
<td>160</td>
<td>Yes</td>
<td>8</td>
<td>120</td>
</tr>
<tr>
<td>Motorola v260/v265</td>
<td>Nov-04</td>
<td>Small</td>
<td>128</td>
<td>128</td>
<td>Yes</td>
<td>8</td>
<td>120</td>
</tr>
<tr>
<td>Kyocera KX2</td>
<td>Nov-04</td>
<td>Small</td>
<td>132</td>
<td>176</td>
<td>Yes</td>
<td>7</td>
<td>125</td>
</tr>
<tr>
<td>Audiovox 8940</td>
<td>Feb-05</td>
<td>QCIF</td>
<td>176</td>
<td>220</td>
<td>Yes</td>
<td>9</td>
<td>167</td>
</tr>
<tr>
<td>LG VX8000</td>
<td>Feb-05</td>
<td>QCIF</td>
<td>176</td>
<td>220</td>
<td>Yes</td>
<td>7</td>
<td>169</td>
</tr>
<tr>
<td>Samsung A890</td>
<td>Feb-05</td>
<td>QCIF</td>
<td>176</td>
<td>220</td>
<td>Yes</td>
<td>7</td>
<td>169</td>
</tr>
<tr>
<td>LG VX4700</td>
<td>Feb-05</td>
<td>Small</td>
<td>128</td>
<td>128</td>
<td>Yes</td>
<td>8</td>
<td>120</td>
</tr>
<tr>
<td>Samsung N330</td>
<td>Mar-05</td>
<td>Small</td>
<td>128</td>
<td>160</td>
<td>Yes</td>
<td>7</td>
<td>121</td>
</tr>
<tr>
<td>Kyocera KX1</td>
<td>Apr-05</td>
<td>Small</td>
<td>128</td>
<td>128</td>
<td>Yes</td>
<td>7</td>
<td>121</td>
</tr>
<tr>
<td>Motorola A840</td>
<td>Apr-05</td>
<td>QCIF</td>
<td>176</td>
<td>220</td>
<td>Yes</td>
<td>8</td>
<td>167</td>
</tr>
<tr>
<td>LG VX4650</td>
<td>May-05</td>
<td>Small</td>
<td>128</td>
<td>128</td>
<td>Yes</td>
<td>8</td>
<td>120</td>
</tr>
<tr>
<td>Samsung A570/A720</td>
<td>Jun-05</td>
<td>Small</td>
<td>128</td>
<td>160</td>
<td>Yes</td>
<td>7</td>
<td>121</td>
</tr>
<tr>
<td>Motorola E815</td>
<td>Jul-05</td>
<td>QCIF</td>
<td>176</td>
<td>220</td>
<td>Yes</td>
<td>8</td>
<td>168</td>
</tr>
<tr>
<td>Samsung A970</td>
<td>Jul-05</td>
<td>QCIF</td>
<td>176</td>
<td>220</td>
<td>Yes</td>
<td>7</td>
<td>169</td>
</tr>
<tr>
<td>Audiovox 180</td>
<td>Jul-05</td>
<td>Small</td>
<td>160</td>
<td>128</td>
<td>Yes</td>
<td>9</td>
<td>151</td>
</tr>
<tr>
<td>Audiovox 8615</td>
<td>tbd</td>
<td>Small</td>
<td>128</td>
<td>160</td>
<td>Yes</td>
<td>9</td>
<td>119</td>
</tr>
<tr>
<td>LG VX8100</td>
<td>tbd</td>
<td>QCIF</td>
<td>176</td>
<td>220</td>
<td>Yes</td>
<td>7</td>
<td>169</td>
</tr>
<tr>
<td>TBD</td>
<td>Aug-05</td>
<td>QGVaL</td>
<td>320</td>
<td>256</td>
<td>Yes</td>
<td>10</td>
<td>310</td>
</tr>
</tbody>
</table>

All dimensions in pixels.

5): Motorola V710 and A840 phones includes a frame around the WAP page reducing width to 167.
Designing for multiple phone groups

Creating elements for multiple device groups isn’t as simple as resizing graphics directly. Larger space may provide room for the insertion of additional text, or iconographic touches that might not fit within their smaller counterparts. Furthermore, while font size roughly parallels screen size, lines of body text that might wrap on one device may not on another. Each handset group should be addressed individually.

In the images above, for example, the different aspect ratio of the smallest size allowed for larger sections of a source photo to be used in the logo (shopping carts), while the promotional image needed to be truncated (“tap for” vs. “tap here for” in the larger images). For that same promotional image, the large version was able to incorporate bolding whereas the medium size required a thinner weight. These may seem like minor touches, but given the small space in which the pages are displayed, they can have a disproportionate impact.

Starting large and scaling down should work for most elements. Some, like icons, will need additional touching-up at a smaller size.
Graphics Guidelines

Tips for compressing images

One theoretical benefit to having limited screen space on which to display an image is that small images require little memory. However, when factoring in the computational limitations of the handsets in question, along with bandwidth considerations, memory does indeed remain in the picture.

Images can be tricky, especially logos. Just as making them larger can cause them to become unfocused, so can radical reduction. If you don’t have a small logo already available, endeavor to use the original font when recreating it, as opposed to simply shrinking the existing logo as a bitmap. If the logo – or any indexed-color image (e.g. a PNG or GIF file) – has an abundance of color, try to simplify it. Limited palettes of 16-20 colors will save space without restricting presentation. Gradients will use up a slew of colors quickly, so use them sparingly. Make sure to contrast the logo with the background, perhaps by adding a border or shadow.

Most handsets currently support 16-bit color – using any more will prove useless.

An important note: do not trust that the image will appear on the phone as it does on your desktop monitor. Handsets feature screens that are not only smaller, but have myriad combinations of color depth, pixel density, gamma correction, brightness, and contrast. An image that appears overly compressed or pixilated on your desktop monitor may prove to be satisfactory on the handset.

Tips for presenting images

For promotional, article, or other photo-realistic images, you may wish to test varying levels of sharpness – using a sharpen or unsharp mask filter – to better display your images on the handset. Be careful: doing so will frequently increase file size. JPG compression can usually be fairly substantial: due to the small displays, anything over 60% compression is usually overkill. To prevent memory bloating, over-compress at first (55% or lower – sometimes a setting even as low as 20 will still be acceptable), test on a handset, and scale back. Progressive JPGs will not display.
As with image compression, the key to image presentation is thorough testing on the actual handsets.

Fonts within images may present problems at first, but once you start collecting a phone-friendly font library you’ll have less trouble. 8-bit fonts such as Silkscreen are incredibly tiny yet legible – perfect for icon titles, etc. For the rest of the graphics, using standard fonts like Tahoma, Verdana, and Helvetica will work well.

The Openwave browser will support animated GIF images, but their large size necessitates the most sparing of use.

Transparent PNGs can be utilized over a `<table>` or `<body>` background image for an aesthetically sophisticated effect, but as with animated GIFs, be very wary of the accompanying increase in file size.

**Heavy image use**

Many customer-targeted web sites, especially content-rich ones and/or sites with heavy advertising, tend to rely upon the frequent use of graphics for both navigation and information presentation. For reasons already covered in this document, this paradigm does not translate well to the phone. While a picture may be worth 1000 words, 1000 words will generally download faster.

However, if you find you must use more than a very minimal number of images on a particular page, MIME multi-part transferring will allow them all to be downloaded with a single connection, cutting down on the deleterious effect that network latency has upon the transference of multiple page elements. This technique is described in great detail in a separate document included in your Verizon Wireless WAP developer package (the Adobe PDF file entitled “WAP 2 MIME Multipart”).
Navigation

You are wedded to the 5-way

We’ve seen how the limited display size of mobile devices severely limits the amount of information able to be presented on a page above the fold, but there is an additional device constraint that affects presentation on the entire page regardless of what can be shoehorned into one screen. The bulk of phones, for the foreseeable future, will rely on a 5-way button (Up, Down, Left, Right, Select) for page navigation. The information placed on a page, especially links and input fields, must take this into account.

The conventional paradigm for interacting with a desktop web page allows for an abundance of links and sizable page length. Shown a complex menu or long list of links, a user with a mouse or other pointing device can simply access their choice, leapfrogging intervening ones. This is not the case with WAP browsers relying on a 5-way button.

The 5-way button requires that the page be scrolled or highlighted link incremented one at a time. Even if multiple links are presented on a single line (outside of tables), pressing the Down key will not advance the highlight to the next line, but instead progress through the current line until no more links are available. Users attempting to follow a link placed far down a page will have to highlight every link from the start of the page to their item of choice.

Fortunately, this situation is consonant with the credo touted throughout this document that WAP pages should be elegant and exacting. Keeping the number of links to a moderate amount will – like minimizing the number and size of graphics – allow you to focus your message and content.
That said, all is not as bleak as one might think. While it’s inadvisable to include 30 links on a single page, a list of around 10 is manageable. In addition, access keys (discussed later) can, if used consistently, help users escape the Sisyphean cycle of repeatedly tapping Down on their 5-way only to come to another page that requires them to do the same.

Another option is to use tables, which allow for more direct movement. Tables aren’t quite appropriate for main body content, but may prove handy for home page navigation or other option matrices.
Pagination of extended content

Just as moving through a sea of links can be a frustrating experience, so can scrolling through a seemingly endless page of text. Without home or end keys, the user is unable to quickly jump to the top or bottom of any given page. Likewise, without an interactive scrollbar, fast and precise scrolling is nigh impossible. The Up and Down keys of the 5-way button will only allow for whole-page movement.

To allow for manageable page size while still presenting lengthy content, consider paginating after three or four screen lengths. Thankfully, WAP 2.0 supports a pre-fetch option using the `<link>` element (see Appendix: Code Samples), so pages likely to be loaded subsequent to the current page can be downloaded ahead of time. This will provide comfortable page size while preventing the annoying delays of waiting for the next page to load upon request.

Reducing the number of links and keeping page sizes small go hand-in-hand. Each is an effort to relieve the user of the onerous task of repetitive key tapping.

User input

Almost all of the elements that compose familiar desktop web forms are available for WAP 2.0: checkboxes, radio buttons, drop-down lists, and text entry fields are all still available. In some respects, forms are even more sophisticated. Given that client-side validation is unavailable to double-check that text input matches a required format, an extension in WCSS – “-wap-input-format” – has been added to allow the browser to enforce formatting rules. See the appendix for an example.

Despite the full range of functionality, developers should remain wary of constructing a highly complex form. Much as navigating through multiple links on a page can be frustrating, maneuvering through a complicated form involving multiple types of interaction (simple Select button presses, text entry) can be more so. If a detailed form is required, your first priority should be minimizing the number of text entry fields. While the –wap-input-format CSS extension will allow
you to restrict the input type (e.g. to numbers for a ZIP code), entry will remain cumbersome.

Drop-down `<select>` boxes will likewise prove difficult if the number of options is too great. A list of all 50 states, for example, may prove unwieldy. Wyoming residents will not appreciate the 49 clicks required to select the home of the Western Meadowlark.

**Access keys**

Access keys allow the quick selection of an element through one of the 12 standard phone keys (0-9, *, and #). When used consistently and clearly – as numbers alongside a static list of home page options for example – they will greatly alleviate some of the clunky obstacles discussed earlier.

Access keys are highly versatile. They can be applied as attributes of an element (e.g. within an `<a>` tag) or as a style class/id using WCSS. Note: if the two are used simultaneously, CSS will take precedent.
Access keys are also contextual. Assigning one to an anchor tag will activate its link, as in the images shown. Assigning one to a form will activate the associated input field (e.g. checking/un-checking a box, or activating a text entry field).

Using access keys, so long as their use is communicated well to the user, is a great way to accelerate site navigation.

Miscellaneous navigational tips

- The headers and footers mentioned in the Layout section above aren’t just a good aesthetic idea: providing quick access from the start or end of a page enables a user to not only avoid having to scroll willy-nilly around a page, but also frees them from having to depend on their browser for site navigation, by relying on the back button or page history.

- The Openwave browser partially supports the WTAI (Wireless Telephony Application Interface) library, meaning that phone numbers can be embedded in the markup and dialed very simply. See the code sample in the appendix for an example.

- Use families of colors to show usability. Using a distinctly colored footer can denote its uniqueness throughout the app. Likewise, similar groups of content can be color-coded for easier recognition. In a simplified UI without the space and graphics to reassure a customer that they are in the right place you must use color to lead people to the right areas, to frame content, to separate the macro from the micro.

- Consider grouping common page types with templates. In our example site, we’ve used the following:
  - A home page – graphically the richest, this page contains an access key-enabled list of links to category pages
  - List pages – these are usually a list of specific content within a sub-area, such as an article. These can be, like any page, several pages long with PREV/NEXT links on the bottom.
  - Content Page – in the case of the examples provided, this is an article, framing the content contextually with any linkable extras. Titles are simplified and/or bolded, dividing lines are utilized for separation, icons highlight photos, pages more than 3-4 page scrolls long are chopped up, a content page footer is placed at the bottom to allow a user to go back to the top, or get more info, PREV/NEXT, etc.
Photo Page – in our example, we’ve avoided putting more than 1 photo on a page with an accompanying caption. Again, icons or simple links to get back to the article, move on, etc. are used.

**Conclusion**

As stated in the introduction, this document is intended to provide a cursory familiarization with WAP 2.0 as handled by current Verizon Wireless handsets. The additional files included in your developer package will explain in further detail some of the more complex aspects of the code and markup.

Focusing your site and adhering to the principles outlined above, you can match or even surpass the desktop experience by creating a tool that places your message, content, and presence in the hands and pockets of a larger base than ever before.

Above all, remember to test your site repeatedly on actual phones. It’s the only way to get a truly accurate picture of technical feasibility and the user experience.
Appendix: Code samples

Embedded telephone number

```xml
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//OPENWAVE//DTD XHTML Mobile 1.0//EN" "http://www.openwave.com/DTD/xhtml-mobile10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <title>Tapi Test</title>
</head>
<body style="background-image:none ">
  <a accesskey="1" href="wtai://wp/mc;12125551212"> 1. Dial</a><br />
  <a accesskey="2" href="wtai://wp/mc;12125551212"> 2. Dial</a><br />
  <a accesskey="3" href="wtai://wp/mc;12125551212"> 3. Dial</a>
</body>
</html>
```

This code will produce three links, each with an access key. Selecting a link, or pressing the associated access key will dial the number 12125551212. The page will appear as to the right.
This code will present an input field that will only accept a five-digit number, after which a hyphen is automatically appended, followed by a four-digit number. Instead of cycling through the alphabetic letters assigned to a key upon depressing it, the number will be directly inputted. E.g., pressing the “2” key will result in an immediate “2”, instead of an “a”, “b”, “c”, and finally “2”.

```xml
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//OPENWAVE//DTD XHTML Mobile 1.0//EN" "http://www.openwave.com/dtd/xhtml-mobile10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <title>Input mask</title>
  <style type="text/css">
    input.zipcode9 {-wap-input-format: "NNNNN\-NNNN"}
  </style>
</head>
<body>
<form action="http://www.example.com/adduser" method="get">
  <p>ZIP code:</p>
  <label>
    <input class=zipcode9 type="text" name="zip" size="10" title="ZIP"/>
  </label>
  <p>
    <input type="submit"/>
    <input type="reset"/>
  </p>
</form>
</body>
</html>
```
Prefetch

Page 1:

```xml
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//OPENWAVE//DTD XHTML Mobile 1.0//EN"
"http://www.openwave.com/DTD/xhtml-mobile10.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<title>Prefetch</title>
<link href="next.html" rel="next" />
<link href="nextbgimage.png" rel="next" />
</head>
<body>
<p>The <a href="next.html">next page</a> and background should load quickly because they're prefetched</p>
</body>
</html>
```

Page 2:

```xml
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//OPENWAVE//DTD XHTML Mobile 1.0//EN"
"http://www.openwave.com/DTD/xhtml-mobile10.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<title>Prefetch</title>
<link href="../style.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-image : nextbgimage.png">
<p>This page and its background should have loaded quickly because it was prefetched</p>
</body>
</html>
```

The <link> tag is used as above to pre-load elements of subsequent pages.
Transparent PNGs

```xml
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//OPENWAVE//DTD XHTML Mobile 1.0//EN" "http://www.openwave.com/DTD/xhtml-mobile10.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Trans Table</title>
  </head>
  <body>
    <table width="100%" cellspacing="0" style="background-image:bg1.png;background-repeat:no-repeat;">
      <tr>
        <td><img src="fg1.png" height="13" width="80" /></a></td>
      </tr>
      <tr>
        <td><br /><img src="fg1.png" height="13" width="80" /></a></td>
      </tr>
      <tr>
        <td><br /><img src="fg1.png" height="13" width="80" /></a></td>
      </tr>
    </table>
  </body>
</html>
```

Transparent PNGs can be used as in this example to overlay graphics on backgrounds. While not recommended for frequent use because of the resultant large download size, it can be a handy way to enhance your layout.