**Mobile Site Information for** [**http://mobile.radisys.com**](http://mobile.radisys.com)

* Only the **essential elements should be brought over to the mobile web**.
* **Figure out what your user would be most interested in** and give them just that
* Keeping rich multimedia to the barest minimum **and** provide alternatives is essential
* Keep the design clutter-free and lightweight
* Mobile web users don’t have access to the traditional keyboard and mouse
* Eliminating graphic elements from your site is usually an effective way to optimize its display on a mobile device.
* Make sure your code validates is as clean and minimalist as possible.
* Give your mobile visitors the option of visiting the standard site
* Limit Scrolling to One Direction
* Use only the images you need to get your message across.
* Many mobile devices have touchscreen interfaces, so try to design with that in mind. That means making the clickable area around your links a little greater, making buttons larger, and putting more space between links.
* No Flash for Javascript
* Include as much of the original site content as possible on your mobile site.
* Make sure redirects work – should you auto forward to the mobile site if you detect a mobile browser?
* When it comes to mobile websites, simplicity is key. Because of the lack of space on the screen and Internet connections that are often slower, it’s important for visitors to have access to what is most crucial
* Include ample white space, this makes the links helpful and easy to use
* Avoid floats and let things ‘stack’ on top of each other: maybe design each section as a stack (like a list) with a small sentence describing each
* The mobile web experience is often a small screen, intermittent, one-handed experience.
* Mobilizing, on the other hand, “precisely targets mobile user needs, making [the] best possible use of technology.” Contextual user tasks — not the existing website — determine the architecture of the mobilized site.
* You need to know your customers, but you need to know what they are likely to be doing on your mobile site, as well as where they'll be when they're doing it.
* The interface of a mobile device is very limited, so if you know what your repeat visitors are coming back for, time and time again, let that naturally bubble up to the top of the site. Avoid burying the content your customers want behind 3 or 4 clicks.
* Showing multiple levels within your navigational listings doesn’t tend to work well because it gives your users too many options to consider and consumes valuable screen real estate. A better way to present navigation is to show only the options related to the page that your users are viewing.
* Having said this, you should always also provide escape points, either as links to the next section, to the parent section, to the home page or all of the above. These links usually work best at the bottom of the page and allow the user to move on, up or out without scrolling back to the top of the screen.
* Designing for the mobile requires switching your thinking to portrait mode where the content is typically taller than it is wide.
* Put the navigation on the first page and on the others pages use a link at the top to come back to the first page or use breadcrumbs.
* On mobile everything is smaller and it is not different with links. You have to make them easier to click on. That can be done with more use of white space around them and bigger font sizes. When the link is selected, instead of just underlining it or changing its color, it should change the background color, even if the link is in the middle of a paragraph. It´s much better for mobile user to see what they have clicked on.
* Each phone has a different screen size, and the iPhone has two positions: vertical and horizontal. So, your site cannot have a fixed width with pixels. It´s much better to use percentages. This method will make your site adaptable to the various screens sizes. Also you can limit the zoom of the interface and the size of the screen using the META properties for mobiles
* Minimize left/right navigation, which is difficult on a phone, and  instead arrange your content in a single column layout
* Make sure that all pages are linked to other pages - It's difficult to get around on a mobile phone, try not to make it  harder by forcing users to hit the back button to escape dead ends

Knowing the users’ devices help web designers create content and experiences specifically for that device

* The Blackberry 9700 uses a display size of 480 x 360 pixels as compared to the Satio’s display size of 360 x 640 pixels.
* The iPhone has a 320×480 pixel screen; the HTC Touch Pro has a 480×640 pixel screen; the Palm Pre has a 320×480 pixel screen; the BlackBerry Storm has a 360×480 pixel screen
* Actual measurements of my Blackberry screen: 192 x 144
* Actual measurements of my iPhone screen: 192 x 240

**Considerations for Mobile Design**

1. Clean, semantic mark-up. The best thing you can do to lay a solid foundation for a usable mobile website is to incorporate clean and semantic markup. What you may be able to get away with on a traditional website may cause significant problems on a mobile website. Clean markup will help ensure that the browser is capable of properly displaying the website, and it will help give visitors a pleasant experience, with no unnecessary difficulties.

2. Separation of content and presentation with CSS Alongside clean, semantic markup is the need for the separation of content and presentation. Mobile visitors are much more likely than desktop visitors to see a website with images and CSS disabled. The most important thing for these visitors is to be able to access the content and links: presentation is secondary. A website that uses clean, valid markup, with CSS to separate the presentation from the content, is off to a great start as a mobile website.

3. Alt tags Because it’s likely that some visitors will not be able to see images on the website (or will choose to disable them), alt tags are extremely important for usability purposes. Of course, alt tags should be used anyway, but it’s even more critical for mobile visitors.

4. Labeling form fields Like alt tags, form field labels help make a website much more usable for mobile visitors. Imagine trying to use a form without knowing what is supposed to go where. Simple details like alt tags and form field labels can make a big difference this way.

5. Use of headings With inconsistent and often limited styling of text on mobile browsers, headings become more significant. Mobile browsers are less likely to style text exactly how you would like it to be, but h1, h2, h3 and other such tags generally help make certain text stand out and build the structure of the page from a visitor’s perspective.

6. Avoid floats if possible Even if a mobile browser correctly displays a website that uses floats for layout, it’s unlikely the website will look good on a small screen. Usually the website will be more usable and look less awkward without floats altogether and with content simply stacked up.

7. Reduce margins and padding Most likely, your mobile website should have smaller margins and padding than your main website has for traditional visitors. Of course, this depends partly on how much of a margin and padding your website currently has, but very large amounts can make the layout awkward.

8. Pay attention to navigation Most websites have a primary navigation menu very high on the page. This is helpful on mobile websites as well, but generally, mobile navigation options are scaled down. Provide only the most relevant links, and, if possible, give visitors an easy way to access the other navigation items.

**9. Consider Color Contrast** Because mobile screens may not have the same appearance as desktop or laptop monitors, make sure the background and text colors provide adequate contrast so that the content can be read easily.

**10. Creating a Simple Mobile IA** The best advice for creating a strong mobile information architecture is to keep it as simple as possible. The following approaches work well when structuring your mobile IA:

Limit choices. Take the content that’s relevant to a mobile user and discard the rest. This results in a simple and focused IA that cuts down the risk of the user getting lost. This approach works well with small, tightly-focused sites.  Create a simple site drill-down architecture that nests content into well-labeled categories. While this sounds straightforward, it’s necessary to plan carefully before taking this approach. A typical Web site has sub-pages: users follow a link -- or "drill down" -- to reach the next page.

**The XHTML Basic Document Type**

You should only use these XHMTL tags for the markup of a mobile site to be safe.

The XHTML Basic document type is defined as a set of XHTML modules. All XHTML modules are defined in the "XHTML Modularization" specification [XHTMLMOD].

XHTML Basic consists of the following XHTML modules:

**Structure Module\***

body, head, html, title

**Text Module\***

abbr, acronym, address, blockquote, br, cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var

**Hypertext Module\***

a

**List Module\***

dl, dt, dd, ol, ul, li

**Forms Module**

button, fieldset, form, input, label, legend, select, optgroup, option, textarea

**Basic Tables Module**

caption, table, td, th, tr

**Image Module**

img

**Object Module**

object, param

**Presentation module**

b, big, hr, i, small, sub, sup, tt

**Metainformation Module**

meta

**Link Module**

link

**Base Module**

base

**Scripting module**

script and noscript elements

**Stylesheet module**

style element

[**Style Attribute Module**](http://www.w3.org/TR/xhtml-modularization/abstract_modules.html#s_styleattributemodule) **Deprecated**

style attribute

**Testing**

Validate that your page conforms to the markup standard you have  chosen using an XML validator like [http://validator.w3.org](http://www.google.com/url?sa=D&q=http://validator.w3.org&usg=AFQjCNGE-DhDRncQHnHci7FWfZo48YINUw).  Further, try using mobile-specific validators like [http://ready.mobi/](http://www.google.com/url?sa=D&q=http://ready.mobi/&usg=AFQjCNHNGQqMSpXyQy7FGNzVnD4ZB4ck7A),  which goes beyond simple validation identifies areas where your page  design departs from generally accepted practices, estimates download  times and cost.

Access the following markup pages on your device to test XHTML and CSS styling:

[**http://cameronmoll.com/articles/mobile/mkp/**](http://cameronmoll.com/articles/mobile/mkp/)

<http://chrispederick.com/work/user-agent-switcher/>

**XHTML Basic**

XHTML Basic is a simplified version of XHTML. It is designed for devices with limited processing power and capabilities such as mobile phones, PDAs, smart watches, pagers, etc. XHTML Basic does not contain XHTML features that are difficult to support on these devices. For example, cascading style sheets, frames, and scripting are not supported in XHTML Basic. XHTML Basic is defined by the [W3C (World Wide Web Consortium)](http://www.w3.org/).

**XHTML Mobile Profile**

XHTML Mobile Profile is the official markup language in the most recent WAP specification version 2.0 defined by the former WAP Forum. The WAP Forum created XHTML Mobile Profile based on XHTML Basic, with the addition of some elements and attributes from the full version of XHTML such as <i>, <b>, <small>, <big> and <hr>. XHTML Mobile Profile supports a simplified version of cascading style sheet called WCSS / WAP CSS.

**WCSS / WAP CSS**

CSS (Cascading Style Sheet) is widely used on the World Wide Web to define how web pages should be presented in browsers. WCSS / WAP CSS is a simplified version of CSS2 with the addition of some WAP specific extensions. WAP CSS is defined in the WAP 2.0 specification. Since WAP CSS is designed for use on wireless devices, CSS2 features that are unsuitable or unnecessary for wireless devices are not included in WAP CSS. WAP CSS enables the separation of the presentation from the content. If you want to change the presentation details of an XHTML MP page, you just need to modify the style sheet. With WAP CSS, you can easily change the layout and style of your XHTML MP pages to suit different user agents.

<http://mobiforge.com/designing/story/comparison-css-21-css-mp-wcss-and-css-level-1>

CSS for button navigation:

<ul>

<li> - css: position relative; width:100%

<span></span> - css: position: absolute; right: 0; bg-img: arrow.gif

<img/> - position: absolute; float: left;

<a href>- position: absolute; float: right;

</li>

</ul>

You need to explicitly declare the viewport, which you do via meta tags. If you declare the width to the constant device-width, it'll default to 320 instead of 980, and everything looks great.

<head>

    <meta name="viewport" content="width=device-width,user-scalable=no" />

</head>

**Development**

* Will anyone else be using this?
  + Do we need an administration interface
  + New products will have to be added manually
  + Product content will come from XML files
  + Solution content will come from
* From PHP, you can grab datasheet through XML files like the first section of the PDF tool
* Use MySQL hold relationship table from ProductID to xID
* Will have to edit from .240 with MySQL Admin tool
* Build series of SQL queries to store and use
* Get Product Category : Product relationships from Site Control (x23?)

**CODING**

XHTML MP documents start with the prolog, which contains the XML declaration and DOCTYPE declaration.

<?xml version="1.0"?> <!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN" "http://www.wapforum.org/DTD/xhtml-mobile10.dtd">

The prolog components are not XHTML MP elements and they should not be closed, i.e. you should not give them an end tag or finish them with />.

The rest of the document is the same as an ordinary HTML document, except that there should be no xmlns attribute for the <html> tag in HTML.

XHTML MP documents must contain the <html>, <head>, <title>, and <body> elements.

**Hello World**

<?xml version="1.0"?>

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

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>XHTML MP Tutorial</title>

</head>

<body>

<p>Hello world. Welcome to our XHTML MP tutorial.</p>

</body>

</html>

!! While 2G i-mode mobiles do play well with XHTML Basic, they do stumble with the optional <?xml version="1.0" encoding="utf-8"?> XML declaration: instead of being a good browser and ignoring this unknown tag, 2G i-mode mobiles render it as plain text.

Although it is a bit naughty to omit the XML declaration if you are serving up anything other than Unicode, doing so allows you to deliver content native to XHTML mobiles that still plays nicely with 2G i-mode mobiles.

#3 Use an XML header and DOCTYPE, and specify character encoding.

<?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN" "http://  [www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd](http://www.google.com/url?sa=D&q=www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd&usg=AFQjCNF-LCSYska3_-gvnH30h_yf_jPVUg)">  <html xmlns="[http://www.w3.org/1999/xhtml](http://www.google.com/url?sa=D&q=http://www.w3.org/1999/xhtml&usg=AFQjCNFx6iI5yL1TV5v1Hh0NXafIXxp4uw)" xml:lang="en">

PHP device detection script:

<http://mobiforge.com/developing/story/lightweight-device-detection-php>

**Avoid tables -** Most 2G i-mode mobiles do not support tables

**Use full URLs with any server-side redirection**

For example, using PHP to redirect mobile users to a mobile site would entail

header("Location: http://example.com/mobile/");

instead of

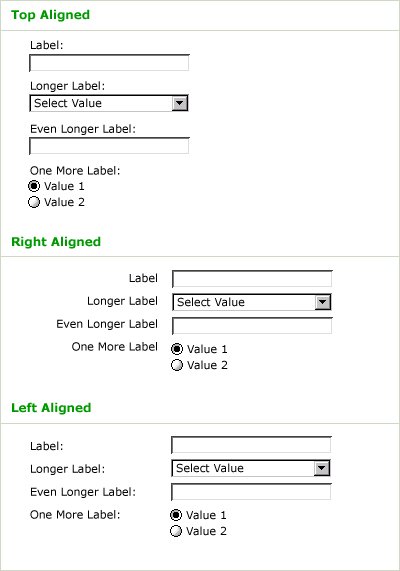
header("Location: /mobile/");

**Forms**

When you do want input from users on mobile devices, radio buttons, checkboxes, select menus and lists tend to work much better than open text fields.

Excessive spacing around the “Submit” button can tuck it behind the keyboard.

Field zoom is another great reason to [top-align input field labels](http://www.lukew.com/ff/entry.asp?504) in forms , left-aligned labels disappear when input fields are expanded to fill the screen.



**iPhone Specific for Email/number/url field types**

It DYNAMICALLY CHANGES THE ON-SCREEN KEYBOARD to optimize for that kind of input.

For example, email addresses are text, right? Sure, but they’re a special kind of text. For example, all email addresses contain the @ sign and at least one period (.), but they’re unlikely to contain any spaces. So when you use an iPhone and focus an <input type="email"> element, you get an on-screen keyboard that contains a smaller-than-usual space bar, plus dedicated keys for the @ and . characters.

<input type=”email” > or “url” or “number”

To sum up: there’s no downside to converting all your email address form fields to type="email" immediately.

**Content and Layout**

**Solutions – Home Page**

* Communications Infrastructure – Level 1
  + LEVEL 2
  + LTE
  + WiMAX
  + Femtocell/Picocell Gateways
  + DPI
  + Mobile Video
* IP Media Servers – Level 1
  + LEVEL 2
  + Audio Conferencing
  + Network Voice Services
  + Enterprise Voice Services
  + Call Centers
  + Voice Quality Enhancement
  + Unified Messaging
  + Video Services
* Military and Aerospace – Level 1
  + LEVEL 2
  + Command and Control Stations
  + Secure Networking
  + Unmanned Vehicles/Robots
  + Portable and In-Vehicle Computers
* Medical – Level 1
  + LEVEL 2
  + Medical Imaging
  + Patient Monitoring
  + Laboratory Equipment
* Embedded Computing – Level 1
  + LEVEL 2
  + Embedded Computing

**Products – Home Page**

* ATCA - ADVANCEDTCA
  – Level 1
  + ATCA Systems
  + Carrier Blades
  + Line Cards
  + Processing Modules
  + Switch Modules
* AMC - ADVANCEDMC
  – Level 1
  + AdvancedMC
* COM EXPRESS
  – Level 1
  + COM Express Modules
  + Carrier Boards
* MEDIA SERVERS
  – Level 1
  + Media Servers
  + Network Architectures
  + Media Server Solutions
* MICROWARE
  – Level 1
  + OS-9 RTOS
  + Virtualization
* MOTHERBOARDS & SBC – Level 1
  + Single Board Computers
  + Endura Q35
  + Endura 945GM
  + Endura 945G
  + Intel 855GME
* EMBEDDED SERVERS
  – Level 1
  + Rackmount Servers
* OTHER PRODUCTS
  – Level 1
  + ENP-2611
  + PCI/ISA Boards
  + Processor PMCs
* Partners
  – Level 1
  + List of Partners
  + Platinum List
  + Gold List
  + Silver List
* About Us
  – Level 1
  + One page with new About Us text
* Support
  – Level 1
  + One page with Service and Support Phone numbers
* Investors – Level 1
  + One page with Investor information to come from Holly

Programming for product pages

* Open up x23.xml(site control)
* Xpath to: /SiteControl/ProductSection/Page[0] to get first level
* Print out all pages: Should be ATCA, AMC, COM-E etc…
* Second Level (IE: ATCA)
  + Mobile/products/atca.php
  + $ProductName = ‘ATCA’
  + Xpath to: /SiteControl/ProductSection[@Name=$ProductName]/Navigation/Page
  + Attach xID to link for opening up that page datasheet.php?xID=5800
  + OR get list from database for productID to xID??
* Third Level – Product Datasheet
  + Image
  + Description
  + Features
  + Specs??
  + Ordering Information??

Bottom Navigation

* Home
* Solutions
* Products
* To fit across bottom in orange links with | separators