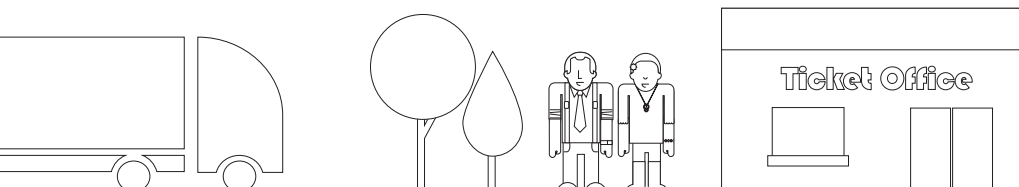




# How to build your mobile site

May 2011



# Introduction

If you have already decided that you want a mobile internet site<sup>1</sup>, this document considers the factors you need to address in building one.

It is important to acknowledge that the mobile arena introduces new layers of complexity that can be difficult for ordinary web developers to accommodate. You not only need to focus on supporting multiple browsers, you also need to recognise the existence of the mobile network operator and the idiosyncrasies that this raises.

When building your mobile site, it is important to identify the scope of handsets you aim to support for the function planned and then determine the number of different presentation layers possible with your budget and the limits set by your desktop CMS. Each presentation layer should be developed in such a way for it to degrade gracefully if the mobile device doesn't support a certain item of functionality. This will allow the mobile site to still be usable independent of device capabilities. This technique is known as progressive enhancement, but alone it doesn't resolve all of the issues you face.

Due to the multiple presentation layers needed, it is important to use an iterative testing approach with regression testing as each new presentation layer is implemented. Establish a project team and devise a roadmap for completion and ensure everyone is aware of developments at each phase of the project and what is required. A new mobile site is likely to generate additional meta data that you will want to incorporate single customer view.

<sup>1</sup> If not, please approach your account manager to see case studies and market research.

# 1 The different development approaches

1.1 As happened with the Internet, organisations are adopting three broad approaches to delivering a mobile optimised internet experience for their customers. Each has advantages and disadvantages and it is important to discuss which approach you want to adopt before going any further:

## Approach 1: Do nothing

- 1.2 As the technology in the mobile arena is progressing at a light speed pace, already many desktop sites are rendering very well and are fully functional on many of the latest, high-end, mobile devices. Improvements can probably still be made to your existing desktop site code to improve appearance and functionality.
- 1.3 The problem is not everyone has the latest mobile device: even in the UK in 2011, more than 60% of the population still relies on feature phones. It's also important to realise that the mobile experience is completely different to the desktop experience and site visitors need to be managed differently depending upon their type of mobile device.
- 1.4 In addition, even for smartphones that can display a desktop website relatively well, the constant zooming and panning is an annoyance and a significant impact to usability and customer experience. If you are serious about entering the mobile market, this should not be the approach to take.

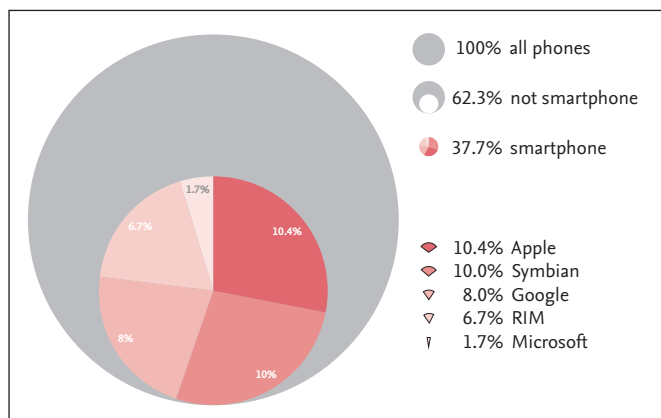


Figure 1: Mobile phones in the UK  
Source: comScore; three months to end February 2011

## Approach 2: Mobile CSS stylesheet within existing CMS

- 1.5 The second approach is to devise a special 'mobile device' stylesheet that can be displayed in place of the existing desktop ('screen') stylesheet within the existing CMS if capable of handling such adaptation. This solution must be hosted by the client, though it has implications for reporting and support/maintenance. This has some significant advantages as it offers the mobile user an improved experience without the organisation having to develop a separate though integrated mobile site.
- 1.6 The problem is the mobile device's detection of the 'mobile device' stylesheet is sparse at best. Some mobile devices don't support CSS or don't handle it very well. It is also only possible to have one mobile presentation layer, meaning you'll deliver a better site to a percentage of your site visitors (e.g. an iPhone presentation layer will address about 10% of all potential UK mobile site visitors), but won't help the much larger majority who will still receive a lower quality experience.
- 1.7 In addition, the mobile site will be exactly the same size (in kB) as your desktop site resulting in poor download times which for mobile is a significant factor.
- 1.8 CSS deals primarily with aesthetics rather than context, often giving little attention to whether the content is even relevant.

## Approach 3: Parallel though integrated mobile site

- 1.9 The optimum approach is to develop and design a specific fully integrated mobile site, taking advantage of the technical features of mobile phones (compared to PCs), and their 'always on and always with us' nature: create content variations (edited versions of PC-content) that are specifically created for mobile browsing, think about navigation designed for different mobile input and access methods and use code specifically written for cross platform support. This offers the best mobile experience and should be the choice for organisations wanting to target the mobile market.
- 1.10 The main disadvantage is the high levels of internal support and resource from the organisation required as this is a new channel and effectively a new site.

- 1.11 You will also need to budget for an initial up-front cost for developing the solution that will provide support for multiple presentation layers.
- 1.12 You are well-advised to consider an integrated solution as this will return significant gains and longer-term cost and effort-savings over developing a non-integrated mobile site.
- 1.13 The parallel option is clearly the best one, assuming the financial, personnel and time resources are available.
- 1.14 This document will focus on the development required to deliver a parallel mobile site.

## Approach 4: Transcoded mobile site

- 1.15 Transcoding is a direct digital-to-digital conversion of one encoding to another. This solution will take the content directly from your current desktop site and present it within your mobile site. This offers a quick and easy alternative to a full integrated solution as content is automatically maintained on the mobile site when your desktop site is updated.
- 1.16 There are drawbacks with this approach; the most significant being any changes made to the coding on your desktop site can break the transcoding engine. The navigation of a transcoded site is restricted as it is required to be a one to one direct mapping of your desktop site. It can also restrict the design as the content needs to be identical, which may have implementations if you want to write mobile-specific copy.
- 1.17 This option is only recommended for an organisation interested in completing a trial period to determine feasibility of implementing a fully integrated solution, or needing, from competitive pressure, to provide a mobile-enhanced version of its website quickly (whilst developing a more involved mobile site in parallel for later deployment).

	Do nothing	Mobile stylesheet	Parallel site	Transcoded
Compatibility	Low	Reasonable	High	High
Experience	Poor	Mediocre	Very good	Good
Content	No change	No change	Changeable	No change
Page size	No change	No change	Much lower	Much lower
Client resources	None	None	High	Low
Cost	None	Reasonable	High	Medium

Table 1: Solution comparison

## 2 Developing your mobile site for cross platform compatibility

- 2.1 When it comes to developing a mobile site, you need to stick to the technical specifications specifically designed for mobile browser support dating back to very early mobile device models from the 1990s when feature phones were being launched. You also need to factor in support for a wide variety of screen sizes and input methods that range from numeric keypads, QWERTY keyboards and the latest touch screen devices.
- 2.2 It is not uncommon for mobile users to disable JavaScript, CSS and particularly imagery from being rendered to improve webpage load speed and possibly to save money if they are on data rate. The solution has to be designed and tested with all three disabled and as such complying with accessibility standards is very important.
- 2.3 Lay down a solid foundation that incorporates clean and semantic mark-up as mobile browsers are less forgiving than desktop browsers.
- 2.4 Separate content from presentation, as content is king on a mobile device and presentation is secondary. Developing to standards defined within Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines (WCAG) should result in a very accessible mobile site solution.
- 2.5 So how do you build a cross platform and cross browser solution capable of supporting the simplest mobile devices to the most advanced mobile devices of today?
- 2.6 The recommended approach is to break the development into stages and create a presentation layer targeting each device classification. As new functionality is added, the mobile site should degrade gracefully so functionality that works on a less supported device is not broken. This method of development is known as progressive enhancement.

## Presentation layers

- 2.7 We believe the following list covers the primary presentation layers required to deliver the best user experience across all mobile devices:
  - tablet
  - hi-resolution smartphone
  - touch screen smartphone
  - keypad smartphone
  - feature phone
  - legacy phone.
- 2.8 Identify which presentation layers you plan to target during the project initiation phase.
- 2.9 But how do you know what presentation layer to display? How can you differentiate between a touch screen smart phone and a keypad smartphone accessing your site?

## 3 Mobile device detection and redirection

- 3.1 It is very important to be able to detect what device someone is using on your website and redirect them to the appropriate presentation layer. If there was no way for a user to go from your desktop site to your mobile site, then it would receive hardly any traffic and there wouldn't have been much point in the investment in the first place.
- 3.2 Mobile Device Detection is often considered a formality but this is still far from commonplace, with some of the largest blue chip companies in the UK failing to redirect mobile browsers to their mobile site.
- 3.3 Detecting whether a mobile device is browsing your desktop site and then knowing which presentation layer and page to redirect that person to is not a trivial process and requires significant thought and development. There are three methods of detection being used:  
**i: No detection, let the user decide**
- 3.4 By simply inserting a link to the mobile site within your desktop site header and/or footer.
- 3.5 The user browsing your desktop site sees the link to the mobile site and clicks it. If you have multiple presentation layers, you will need to have a link for each.

- 3.6 The benefits of this solution are quite clear – it is very quick and easy to implement at very little cost. The disadvantages are:
- Mobile users have to first wait for the desktop site to load (probably slow, displays idiosyncratically, difficult to navigate once loaded).
  - They may not see the link to the mobile site (at least put it at the top).
  - The link to the mobile site may not fit in with your current design.
  - Multiple mobile layer links require clear and accurate naming (that makes sense to the site visitor).
  - Multiple mobile layer links are probably not going to provide a good site visitor experience – and both mobile and PC visitors will be exposed to this strange site choice).
- 3.7 Improvement to the user experience could be included by using cookies to detect what presentation layer the user selected first time round and upon their next visit they are automatically redirected.
- 3.8 This does require the mobile device to be capable of supporting cookies which not all of them can, and of the site visitor being mentally prepared to go through the effort of loading your site on their phone a second time.

## ii: JavaScript detection

- 3.9 JavaScript is a client side technology that can be used to implement the redirect. The problem is identifying whether the device is mobile or not and what type of mobile device it is, and then keeping this script up to date as new mobile devices are released (typically 50 new devices per month). A more significant issue is that many of the older

browsers on mobile phones do not support JavaScript or have JavaScript disabled by default. Many users also prefer to disable JavaScript to make browsing faster. JavaScript detection is by no way an ideal solution.

## iii: User agent detection

- 3.10 This solution is by far the best way to detect and redirect the user to a mobile site. Every device that accesses a webpage sends over a HTTP header that includes a user agent description. Detection can then be achieved by cross referencing each of the user agent strings accessing your desktop site with a database of mobile user agents to determine whether it is mobile or not and what type of mobile device. The detection and redirection is implemented server side so it is not reliant on the capabilities of the mobile device in any way.
- 3.11 The optimum approach is to utilise an online database of user agents that is regularly kept up to date. You can then connect to this database using a web service when a new session on your desktop site has been created. This solution offers the guarantee of the database being up to date and doesn't require any manual intervention.
- 3.13 Deploying a User Agent detection script that uses an online database via a web service is the recommended approach. It is relatively straightforward to implement and needs to be added to every page of your desktop site, but with common desktop profiles cached.

## Option to revert back to desktop site

- 3.14 Depending on your detection approach and content and functionality available on your mobile site, it

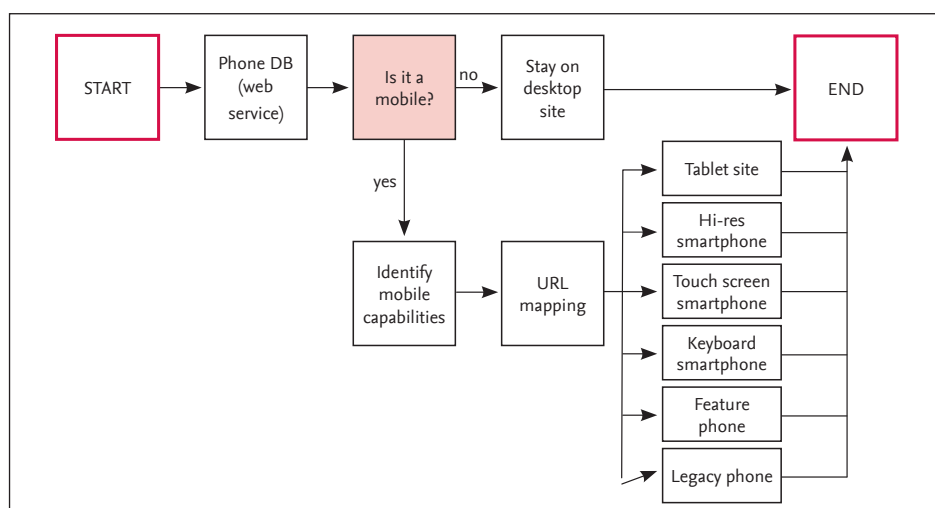


Figure 2: Mobile site re-direction

Incentivated has a **Phone Capability Database (PCD)** which contains thousands of handset profiles and is updated dynamically each day when new firmwares are released, ensuring that we are up-to-date in advance of consumer release. We receive regular updates from the manufacturers on new model launches, monitor the information for inconsistencies (based on all mobile traffic over our servers) and manually test as many handsets as is feasible.

is always important to include a link back to the desktop site from your mobile site.

- 3.15 If you have implemented an automated redirection approach, then it is important that when the user session has been redirected back to the desktop site for the desktop site to not automatically redirect them back to the mobile site. To get out of this infinite loop, you need to include an override attribute within the redirect URL from the mobile site to the desktop site that can then be checked.

## Redirection comparison

- 3.12 Comparison of the three options is as follows:

	No detection	Javascript detection	UA detection
Scope	None	Small	Large
Maintenance	None	Medium	None (if using the web service)
Implementation	Small	Medium	Medium

Table 2: Redirection comparison

## 4 Mobile site redirection

- 4.1 Once you have detected that there is a mobile device accessing your website, you now need to redirect them to the appropriate web page on your mobile site. Several companies have just taken the approach of redirecting all mobile traffic to their mobile site homepage. If the person is navigating to the site by using a deep link URL from a search engine or external link, having to navigate down to that deep content from the home page offers very poor user experience.
- 4.2 Depending on your implementation choices, the following redirect options are available:

### URL modification

- 4.3 If your mobile site is an exact replica of your desktop site and the URL structure is the same apart from the domain name. The simplest approach is to modify the URL by only rewriting your domain name to be that of your mobile site's domain name.

### URL mapping

- 4.4 URL modification is not normally possible as the mobile site can be a subset or completely different navigational structure to the desktop site having been optimised for mobile browsing.
- 4.5 In this case, a URL mapping solution needs to be deployed where the URLs on the desktop site are

mapped to equivalent URLs for your mobile site. If there is no equivalent URL on the mobile site, then it is important to default to the homepage (or section header page) and not an error page.

- 4.6 It is important to keep this URL mapping solution up to date as new pages are added to the desktop and mobile site. A script could be specifically designed to simplify this process. In the case of a 404 "page not found" error on the mobile site, this should then redirect the user to the homepage. In the case of an HTTP 500 error response, then display the mobile site's standard error page.

### Common broken site link errors

301 error = moved permanently  
 400 error = bad request  
 401 error = unauthorised  
 404 error = not found  
 500 error = internal server error

## 5 Mobile site coding standards

- 5.1 There are four coding standards that are used to build mobile sites.

### i: WML 1.0 (wireless markup language)

- 5.2 In the early days of mobile web, mobile devices were technically very basic and the screens were low-res and not in colour. A special mobile web mark-up language was created to cater for displaying web pages capable being seen on what we now call 'legacy' mobile devices.
- 5.3 This mark-up language was called WML 1.0 which is an XML mark-up language based on the card-and-deck metaphor. It could be used by devices that implemented WAP (Wireless Application Protocol).
- 5.4 Due to the basic design and lack of functionality, the adoption rate for a WAP 1.0 mobile site was not very high. It was also very expensive and slow in the early days to browse the mobile web on your mobile device (first and second generation mobile networks were not very capable of transmitting large amounts of data).
- 5.5 You should only really be considering building a mobile site in WAP 1.0 if you are specifically targeting legacy phones (Nokia 1100 and 1101) which are more prevalent in some developing nations.



## **ii: CHTML / i-mode-HTML (compact hypertext markup language)**

- 5.6 At the same time as WML was released, CHTML was designed as a subset to the HTML mark-up language. This made it easier to develop than the WML specification and was adopted by the Japanese market with huge success.
- 5.7 CHTML was a more advanced solution to that of WML and resulted in a significant uptake of mobile web usage within the Japan market. Mobile web usage in Japan is still the highest out of any country to this day and is seen as the leader in developing and adopting mobile technology throughout the world.
- 5.8 Some of the key advantages of CHTML were the addition of functionality specifically targeted at improving the mobile experience. This included access keys and phone number shortcuts.

## **iii: XHTML-MP / WML 2.0**

- 5.9 As the mobile devices evolved and colour screens started to become the norm, WML 1.0 was no longer suitable, and neither was CHTML. Site visitors wanted a richer experience reflecting their expectations from browsing the Internet on their desktop PC.
- 5.10 WML 2.0 was born out of taking a subset of the XHTML 1.0 specification, but unlike CHTML and WML, it included support for images and CSS. Unfortunately, the name WML was too closely aligned with WAP 1.0 and mono websites, so the specification was named XHTML-MP (external hypertext markup language - mobile profile).
- 5.11 XHTML-MP is so similar to the XHTML specification that many mobile sites are developed using the XHTML doctype instead. Modern day web browsers on mobile devices are now more than capable of supporting full XHTML specification.

## **iv: HTML 5**

- 5.12 The World Wide Web has now become a mismatch of websites following different HTML standards with each having different advantages over each other and all being implemented poorly on various occasions.
- 5.13 HTML 5 was created to consolidate the best features of all the different types of HTML and to extend and improve on these existing features. It is now a far more capable solution offering new syntactical features and support for audio and video content.

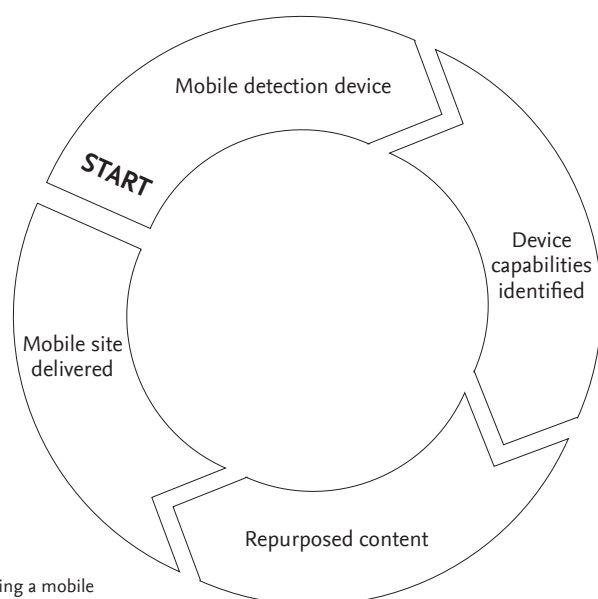
But why use it to build a mobile site when most mobiles don't support anywhere near the full set of HTML 5 features?

- 5.14 There are reasons. Firstly, HTML 5 is an extension to XHTML specification, and as such HTML 5 functionality can be used in a progressively enhanced way to build mobile sites that are compatible with older devices. Using HTML 5 will future-proof your solution for many years to come.
- 5.15 Secondly, the success of CHTML was due to its ability to deliver key usability improvements for the user. The HTML 5 specification has taken these improvements into account and has added even more. It is a far more capable solution and will be able to deliver a considerable improvement to mobile sites that have been developed using XHTML and any prior incarnation.
- 5.16 Some of these improvements include:
- Speed improvements:
    - Using the new web browser storage database to store a large amount of data instead of the limited storage capabilities offered by cookies.
    - Improved offline caching support.
    - Synchronous JavaScript calls.
  - Form field improvements:
    - Built in RegEx provides validation of email address formats and input fields only requiring numerical values.
    - The mobile browser can update the virtual keyboard to display only numbers if the field is of type 'number'.
    - Geolocation capabilities to access the mobile devices location.
- 5.17 However, you can't rush in today and start using all of the features HTML 5 offers if you want your mobile site to be accessible by the majority of mobile phones. You still need to clearly think through what features you want to provide and can support and how to go about implementing them for all, or as many of your site visitors as possible.



## 6 Design to the phone's capabilities

- 6.1 At the start of the document, we mentioned that progressive enhancement techniques should be used when building your mobile site. We also mentioned that a progressive enhancement technique is not able to deliver a fully optimised mobile site offering the best user experience across all mobile devices.
- 6.2 Let's start by identifying what progressive enhancement cannot deliver:
- You do not know the resolution size of the mobile device's screen resulting in any image not being displayed accurately. For example, if you create all your images to be 240 pixels wide, the image will appear compressed on small screens and distorted on larger screens.
  - You do not know what types of images your mobile device supports. Some mobile devices do not support GIF and older phones do not support PNG. There is a similar issue with which video format you choose.
- 6.3 To resolve these issues you need to know about the devices accessing your site before the web page begins to render. You need to have access to a database containing the capabilities of all mobile devices and use a server side script to dynamically generate or switch between the various solutions to optimise your site. This database should be updated daily.



## Accommodating different screen sizes

- 6.4 A mobile device (excluding tablets and some newer smartphones which can rotate their displays) is horizontally challenged and as such mobile sites should be designed to allow the user to scroll through the content vertically and not horizontally. The challenge is building a mobile site that avoids the user having to pan, zoom and scroll horizontally to see the content, but also changes the viewpoint when rotated by 90°.
- 6.5 Unlike a desktop site where you design and build the site to work on either 800 or 1024 pixels width, the widths of mobile devices varies from 76 to 480 pixels and it is important to fill the full width. Coloured blocks can and should be developed using CSS to offer separation and improve download speeds.
- 6.6 But for images, each image needs to be generated for each of the various screen resolutions. The difficulty is there are already 10s of screen resolutions with new screen resolutions being launched every year (generally increasing in size).
- 6.7 But it is important that if you change an image, that new image needs to appear in place of the old for each of the different screen resolution layers.
- 6.8 The screen size needs to have initially been obtained from a mobile device capabilities database that has the information on the screen resolutions of every mobile device.
- 6.9 However, what you do not want to do is repeat this task every time a mobile device of the same screen resolution requests the same image. Every time an image is created, it needs to be cached and for each subsequent request, the cache should be checked to see whether it already exists (this increases page load times as well). If it doesn't exist then the image should be recreated. If you change an image then you need to delete the image cache so new images can be re-created for each of the different screen resolutions.

## Device capabilities identification

- 6.10 Mobile devices can have different OS versions and different web browser versions resulting in a very complex mesh of different capabilities that can be supported (over 10,000 combinations). Common issues include knowing what images to display, what video can be rendered, whether flash can be used and does the mobile device have a touch screen.

- 6.11 The answer to all these questions is having access to a phone capabilities database that already knows the full range of capabilities of each device. It will then be possible to apply server side scripting technology to dynamically alter the presentation code to optimise for each device. As with the database required to dynamically repurpose image content, this database needs to be kept up to date as and when new mobile devices are launched onto the market.
- 6.12 But it doesn't have to just stop at what image or video formats your device supports.
- 6.13 Let's say you have developed a complex application online targeting at mobile devices and you want the user to use access keys to navigate around the solution. So you design a help section showing what each of the access key does. But what if the mobile device has a QWERTY keyboard or a non-standard keypad? Using the device capabilities database, you can devise a solution catering for each and switch between them. For mobile devices that have a touch screen, this option might be removed altogether.
- 6.14 You can now certainly see the potential of devising a very customised and personalised solution online. The opportunity to optimise your solution to cater for all mobile devices is now possible.

## Additional optimisation techniques

- 6.15 Bandwidth is critical in any mobile discussions – especially when it comes to designing your internet site. Until next generation (4G) mobile networks arrive, and/or wider availability of wi-fi hotspots for accessing data, any mobile site you create needs to be optimised so that it has a small data footprint and can be quickly downloaded. Each mobile site page should be no larger than 30kb for it to work on all mobile devices.
- 6.16 The following are optimisation techniques that can be made to your site to reduce the download time.
- **Use only one JavaScript and CSS**
  - **Compress JavaScript and CSS files**
  - **Consolidate your CSS and JavaScript files**
  - **Avoid using images if possible, and instead use the full capabilities of CSS**
- 6.17 It is also important to optimise the user experience of the mobile site. With this in mind, some of the usability improvements should be:
- **Make telephone numbers clickable**  
As the mobile web is dominated by mobile phones, you can make a phone number on a webpage clickable to make it easier to make

a direct call. This is done by adding an HTML link to the phone number, and there are several standards for this. You need to use a mobile capabilities database to know which one to use. In addition, the link needs to be formatted with the International dialling code.

- **Always define image dimensions**

Specifying the dimensions of each of the images will allow the browser to download text first and leave the correct space for the images while they download. This avoids the reader having to jump around the page as images are loaded.

## Things to be aware of

- 6.18 We have covered a lot of the detail involved in building and optimising your mobile site, but there are more traps to be aware of if you don't (or can't) stay up-to-date and aware of the advancements in technologies (and mobile technologies are moving incredibly quickly at the moment).

- **Opera Mini**

The first item to bring to your attention is the Opera Mini mobile browser that renders pages on the server and sends a snapshot of the page to the device. Effectively the user downloads an image each time and this image is also cached on the Opera Mini server.

- **Flashlite**

Yes, there is a Flash version specifically designed for use on mobile devices which is more lightweight to the full desktop flash version. But, Apple (and therefore the iPhone) does not support or work with Adobe Flash, and so iPhones do not support flash (as standard). If you want to use Flash to display video or animation then it is vital to provide an alternative.

- **Font Support**

When it comes to embedding fonts, iPhone's Mobile Safari browser doesn't fully support @font-face with preference for SVG (scalable vector graphics) file format. However SVG is not fully supported on a range of web browsers including Firefox and IE so you will need a hybrid approach to target all browsers.

# 7 Implementing your mobile site

7.1 When you develop a desktop site, one of the biggest challenges is making the site work across all web browsers; particularly the older editions of Internet Explorer. For mobile, there are an even larger number of mobile browsers combined with the issue of supporting multiple platforms each with different capabilities and different technological support. Not only is there a significant technical complexity, there is also the long term strategy for content maintenance and support for the site moving forward.

7.2 There are three approaches when devising a solution design to support a mobile site:

## i: Independent standalone mobile site

7.3 A standalone mobile site without any integration is by far the cheapest and quickest to deploy. It is recommended that the mobile site is also run from a CMS to manage and update the content as agency fees for content maintenance can quickly add up.

7.4 The biggest disadvantage is the requirement for double keying (you have to write the same copy twice – or copy and paste between two CMS’ – leading to increased likelihood of errors and/or mobile and desktop sites having different/contradictory content) if the content on your mobile site also needs to contain the same content as your desktop site.

## ii: Integrated mobile site

7.5 An integrated option offers the optimum solution to allow the site to be designed and developed specifically for mobile with the added advantage of content and functionality integration. This will reduce the internal maintenance effort for the mobile site. Integration is usually completed with your existing CMS and can be done in one of two ways.

### a: Direct integration with your CMS (client hosted)

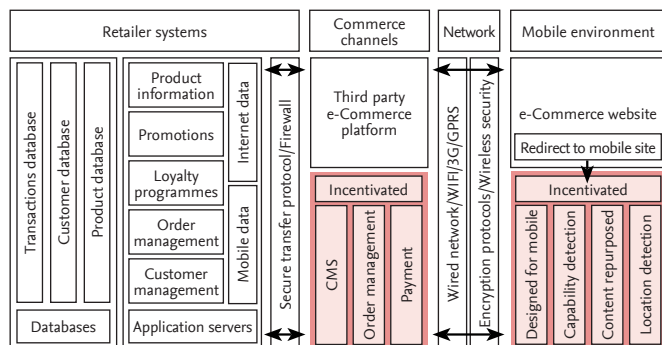
7.6 Your CMS will need to have support for multiple templates. If it doesn't, then this is not a viable option. Specialised mobile templates can be created and included within your existing CMS and integration with back office CRM systems can then be factored in with web service calls or direct SQL queries.

Advantages:

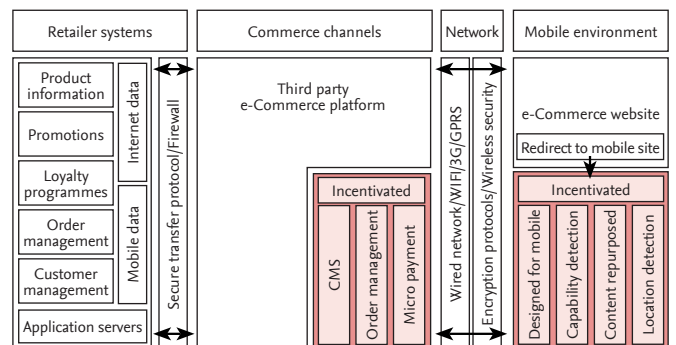
- Static content can be managed and maintained by existing processes as used for the desktop site
- Hosted within the client's existing web environment so no additional hosting costs will be incurred
- Organisation will take ownership of the code

Disadvantages:

- External maintenance of the code may be required by the 3<sup>rd</sup> party agency and will require remote access which may be a security risk
- 3<sup>rd</sup> party agency's warranty is voided if any changes have been made to the code if the 3<sup>rd</sup> party didn't complete/authorise the change.



Approach 1: back office integration



Approach 2: Integration with an eCommerce platform

## b: Integration with 3<sup>rd</sup> party platform (externally hosted)

- 7.7 Even though direct integration with your CMS seems the best solution, due to the complexity of supporting a cross platform architecture, a specialised platform for rendering mobile sites may be a better alternative – particularly if your CMS is only capable of supporting one presentation layer for mobile.
- 7.8 Mobile technology is evolving very quickly and making use of a mobile platform to deliver a site can future proof your solution for any new technology that may be around the corner.
- 7.9 The integrated approach can be achieved using either a content feed or web services from a CMS.
- 7.10 **Content Feed:** The CMS will be required to produce a feed of content in an agreed format (XML or JSON etc.). This feed can then be updated nightly, weekly or upon a change to your desktop site and transferred across to the 3rd party platform.
- 7.11 **Web Services:** Web services can deliver static and dynamic content and can offer improved functionality to the mobile site by direct integration with an organisation’s back office systems.

### Advantages

- Made for mobile functionality
- Code maintenance is outsourced
- Future proofed solution

### Disadvantages

- Code is probably not interoperable between platforms

## iv: Combination of the above

- 7.12 It can be the case that your mobile solution can be created out of a combination of the above three options, or all three supporting different sections of the mobile site. This is ideal for an organisation wanting to adopt a phased approach to the mobile strategy.
- 7.13 For example:
- Your top level pages (Homepage, About Us page) could be standalone giving you flexibility to create and design the layout and content.
  - The product pages and a forum could be transcoded. This will allow for any new products or forum posts to instantly appear on your mobile site.
  - A user signup and login will require an integrated solution to connect to your CRM to create a new account and authenticate user login process.

## Conclusion

- 7.14 Deciding on which approach to adopt is dependent on your requirements and your customer’s mobile behaviours and device preferences. An external party can assess your current solution for feasibility of the different options and provide a recommended approach.

	Standalone	Transcoded	Integrated
Speed to deploy	Quickest	Quick	Slow
Content maintenance	Manual	Automated	Automated
Mobile site control	Fully editable	Restricted	Fully editable
Quality of design	High	Mediocre	High
Longevity	Customisable	Restricted	Customisable

Table 3: Integration comparison

## 8 Approach to testing your mobile site

- 8.1 Development of your mobile site is normally done using a standard desktop PC, but testing using your desktop web browser can only get you so far in terms of simulating the mobile experience. There are many elements of mobile device usage that can't be replicated accurately in this way.
- 8.2 One of the most concerning issues is a mobile operator might restrict data packet sizes to something smaller than you expected resulting in your images or even your web page not loading, or taking much longer than is acceptable. Images may be compressed when accessed over cellular.
- 8.3 It is important to acknowledge that mobile device web browsers are more often than not significantly less capable than a desktop web browser and mobile devices have fewer hardware resources than a desktop PC.
- 8.4 It is also important to test whether your dynamic content repurposing functionality works for the different mime types that are supported so that JPEG images are rendered if GIF imagery is not supported.
- 8.5 There is no limit to the amount of testing you can do for a mobile site, but it is clearly not feasible for an organisation to test their mobile site on every mobile device in existence (1,086 different phones are listed as being in use in the UK in January 2011, by comScore, for example). There are alternatives:

### Emulators

- 8.6 There are numerous emulators available online provided by the mobile device manufacturers or developed independently. These allow you to quickly see images in context and the general layout, but they are not real devices and are liable to have their own quirks and differences. They should only be used as a first pass and not considered a substitute for the real thing.

### Remote access to mobile devices

- 8.7 There are services available that allow remote access to a mobile device allowing you to control its features and test a mobile site. This service does cost, but it would be significantly cheaper than purchasing lots of different phones. For a simple website, this is probably more than is needed.

### Buy a collection of devices

- 8.8 Buying a subset or a representative sample of devices you want to target is a good solution. You probably want to focus on purchasing the most commonly used devices (a top 20 of those used by your customers<sup>2</sup>, for example) and consider a device from each of the major manufacturers.

### Ask your colleagues and friends

- 8.9 If you build a one-off mobile site then this is probably the best and cheapest way to test to confirm the solution. Hopefully you know enough people that have a significant variety of phones that make this approach feasible.
- 8.10 The scope of testing should be based on the scope of presentation layers that are being targeted. There is no point testing with a WAP 1.0 phone if you do not have a presentation layer supporting WAP 1.0. Similarly, there is no point testing on a tablet device if you plan on redirecting all tablet devices to your desktop site.
- 8.11 As well as the above, it is important to validate your code mark up conforms to W3C standards. <http://validator.w3.org> is the recommended tool for validating your code.

2 A research survey like comScore can provide a detailed breakdown of primary handsets used by a variety of socio-demographic filters.

## 9 Statistical reporting

- 9.1 Finally, once you have completed building your mobile site and it is live, the business will want to know how well it is performing.
- 9.2 Your current desktop website analytics tool is probably not tracking all your mobile site traffic. A good desktop analytics tool may only be 50% accurate – cheaper ones significantly less than that!
- 9.3 You cannot rely on desktop analytical reporting tools; you need a specialised mobile analytics package. This is one of the biggest challenges when making a decision as to whether you need a mobile site in the first place. Let's say you are using Google Analytics and it tells you that 5% of your desktop traffic is mobile so you think you should investigate the feasibility for a mobile site. In reality it is probably double that or more.
- 9.4 The issue with existing desktop web analytics tools is that they use JavaScript and cookies to provide the reporting. As mentioned above, JavaScript should not be relied upon and some mobile devices only support a subset of the functionality JavaScript offers, if at all. You also have to consider the multitude of custom made mobile web browsers that include Opera Mobile and Skyfire. Your analytics package may or may not consider these be a browser for mobile devices.
- 9.5 In addition cookies don't always work and the IP address of a mobile device is the mobile network's IP address, not the device itself. Measuring uniqueness amongst your mobile traffic now becomes a significant challenge.
- 9.6 The solution is to use a server-side script to track and report on mobile visitors, combined with the aforementioned mobile device database to be able to know what mobile devices are browsing your site. Cookies should not be relied upon for tracking uniqueness.
- 9.7 A complete mobile analytics solution will obtain a unique identifier from the mobile networks that in most cases is the user's mobile phone number encrypted. This should be used for measuring uniqueness and will be far more accurate at measuring unique visitors than a desktop analytics package which relies on IP-addresses and cookies. This normally requires close relations with the mobile networks to obtain, and there are lots of mobile networks globally.

Your analytics tool probably shows that there are only the latest mobile devices browsing your site on the basis that it can't track the older phones, not because the older phones aren't visiting your site!



## Conclusion

As the worldwide shift to mobile continues, mobile device support will become increasingly important. One of the major issues is that for older devices there is no clear upgrade path for them to move to the latest software so you will always need to be backwards compatible. This is unlike desktop browsers where reminders are sent to prompt upgrades and upgrading is made so much easier through easy-to-follow guides through official upgrade web pages.

We hope that this paper has left you with a clearer understanding about the different approaches to developing a mobile site there are and the issues you need to be aware of.

It is important to be aware that the technology for a mobile site is still in its infancy. There are new libraries for mobile sites including touch detection for example, and best practice approaches to building a mobile site change year on year.

Just as new standards have emerged for desktop sites in the past decade, new standards are emerging to unify mobile browsers.

What this means is that techniques described in this article are only temporary and if you are developing and owning your mobile site you should attempt to stay on top of this ever changing mobile roller-coaster.

If this is not feasible within your organisation, there are plenty of specialists in the mobile arena to assist in developing and implementing your mobile strategy.



## Contact us for more information

Email: [info@incentivated.com](mailto:info@incentivated.com)

Web: [www.incentivated.com](http://www.incentivated.com) or scan the QR code below to see our optimised website on your phone.

Tel: +44 (0)20 7392 2323

A selection of other white papers we offer:

- mCommerce
- Mobile coupons

Incentivated is an independent technology company with 10 years' experience operating exclusively in the mobile marketing services sector.

We help our international client base engage with their customers by designing, developing and delivering integrated acquisition, retention (CRM) and transaction (mCommerce) campaigns and services for mobile.

Our proprietary technology and specialist staff are well positioned to help brands, the public sector and charities to develop everything from enterprise messaging (SMS & MMS) through to mobile internet sites, to server-side software or handset applications, including web-apps, for 'smartphones' and feature-phones.

We also provide strategic, creative and technical advice for the use of mobile by businesses to raise awareness, deliver marketing ROI and provide customer service.

Scan the QR code below to see our website optimised for your mobile phone, but accessed through our existing website URL.

WP-M 1.5/05-11

