



Developers' Guide

Framework

Version 2.0.3.1

20th January 2012

© 2011 51Degrees.mobi Limited. All rights reserved.

The copyright in and title to the document "Developers' Guide – Framework" belongs to 51Degrees.mobi Limited. No part of it whatsoever may be reproduced in any form without the prior authority of 51Degrees.mobi Limited and/or any original source as appropriate. Any agreed copy or extract must be marked with all proprietary notices which appear on the original and will be subject to the requirement that you will acknowledge on the face of each part of the reproduced material that it belongs to 51Degrees.mobi Limited.

3 Introduction

This User Guide has been written for those that have a passing knowledge of ASP.NET development in Visual Studio and wish to create a compelling mobile web application customised for the mobile device. In many cases we're building on work done by Microsoft with ASP.NET and providing developers with options to improve productivity and simplify developing for mobile web applications. Where we are extending or building on concepts already covered extremely well within Microsoft's documentation we've provided links to the relevant pages on the MSDN web site either within the document content or as suggested further reading.

If you've created a web site before, we'll explain some of the differences you need to consider when working with the small screen of the mobile device. A mobile is a very different beast compared to a desktop or laptop. Nothing can be taken for granted concerning supported standards and technologies. Mobile device screen size and resolution vary extensively. Screens vary in size from 100 pixels square to 640 x 960 pixels on the highest end smart phones. The growth of tablet mobile computing devices such as the Apple iPad increases the variations. As higher resolution screens come to market the pixel size will become less important than the physical dimensions. The ubiquitous keyboard and mouse/touchpad have been replaced with a dazzling array of varied input methods. Before continuing be prepared to think differently about design.

This chapter introduces the reader to the overarching principles of the 51Degrees.mobi product set. It does not cover any implementation details and unlike subsequent chapters is authored with a broad audience in mind. By the end of this chapter the reader should understand the design and technology concepts that underpin the 51Degrees.mobi product set and gain a broad understanding of the features available.

The chapter starts by exploring the design concepts behind 51Degrees.mobi, before exploring the technical concepts.

3.1 51Degrees.mobi Design Concepts

Members of the 51Degrees.mobi team have been developing mobile web applications since early 2000 with large corporations, small businesses or as individuals. Our experiences have taught us how to overcome hundreds of common technology and design challenges associated with developing for mobile devices. With 51Degrees.mobi we set out to solve these challenges in one place, creating an extension to ASP.NET that would benefit everyone. Importantly we want to consolidate our experiences in a single product enabling others to avoid reinventing the wheel or needing to become pioneers.

51Degrees.mobi isn't just about mobile phones. We're about all mobile devices including tablets, phones, watches, remote controls, and anything else that can be carried around and gets connected to the rest of the world via a radio data connection. We don't mind if it's got a SIM card in it, uses GPRS, 3G, LTE, WiFi, WiMax, or any future technology. That's why we refer to "mobile devices" as a term to encompass any device that can access a mobile web application.

This section explains the design concepts that have been employed and sets the scene for the technology concepts explained later in the chapter.



3.1.1 Single Code Base = Better Business Model

The popularity and pervasiveness of the Windows desktop operating system has enabled application developers to capture over 80% of the market using a single application written for Windows alone. Many application developers

have never needed to consider Apple or Linux platforms to enjoy a profitable business model.

Native mobile applications are written using the bespoke framework of the mobile platform manufacturer and are not portable across platforms. An application written for the iPhone cannot easily be ported to Android without requiring a complete rewrite. As no single mobile platform yet enjoys the dominance of Windows on the desktop, application developers targeting the features of native applications will need to produce many different versions if they wish to target the widest possible customer base. This approach can be costly and time consuming often making such an application unprofitable.

Software as a Service (SaaS) has enabled web applications to be created and deployed anywhere a suitable desktop browser has been available. 51Degrees.mobi uses the browser on the mobile device to deliver the application. Many devices and platforms can support a single mobile web application developed using a single code base and business logic. The 51Degrees.mobi User Interface (UI) will automatically adapt to support the mobile device and browser being used from a single code base. Smart phones and feature phones can all be supported with a single application.

51Degrees.mobi supports all the major mobile device manufacturers including Android, Apple, Blackberry, Palm and Nokia.



Figure 1 - Supports all major mobile devices

3.1.1.1 Date Example

Many applications will at some point require the entry of date information. Restrictions may be required limiting the dates that can be chosen. For example; weekends may be invalid, or dates that have already been fully booked should not be available. The overall business process may need multiple dates. For example; a start and end date to specify a range to be used with a reservation service.

The UI design employed to enable the user to provide this information can vary greatly depending on the capabilities of the mobile device. The 51Degrees.mobi calendar control will automatically adapt to different mobile devices.

In its simplest form the control will enable dates to be entered using drop down lists positioned one under the other to specify day, month and year when applicable. See Figure 2 - Calendar in Drop Down List Mode.

Day	23 ▾
Month	Oct ▾
Year	2010
Select	

Figure 2 - Calendar in Drop Down List Mode

This input method works well on devices controlled via joystick or trackball with narrow screens incapable of displaying multi column tables. However many discrete inputs will be needed to provide a date and select it.

With touch input methods a more intuitive and easier to use calendar can be presented requiring fewer inputs to select the date(s) required. See Figure 3 - Calendar in Grid Mode.



Figure 3 - Calendar in Grid Mode

In grid mode it's possible to provide two dates in the same month with only two presses of the screen. This approach is significantly quicker than drop down lists. Further if the mobile device supports gestures the month can be changed with the swipe of a finger through the calendar horizontally.

51Degrees.mobi controls ensure an optimum UI is created for the mobile device utilising a single common code base and business logic reducing development costs and improving the business model.

3.1.2 Keeping Mobile Simple

Businesses have invested heavily in the technologies they use. This investment is not just in the licence fees and the hardware deployed but in the skill sets of individuals. 51Degrees.mobi has been designed with the following overriding requirements in mind:

1. No new hardware.
2. No new skills.



A web site's current hardware is likely to be supporting its current demands. If not this should be addressed before considering mobile! For many businesses a mobile web application will result in an increase in traffic as the services provided can be accessed more easily. However, there will be some cannibalisation of traditional web traffic as users fulfil their requirements via mobile devices instead of desktop browsers. Most web sites will already have given some thought to capacity planning and have plans in place to provide additional web servers or other solutions. 51Degrees.mobi builds on this existing investment and thought by enabling the mobile web application to be built alongside the existing web site using the same hardware and technology. 51Degrees.mobi facilitates the creation of additional web pages designed and optimised for mobile devices. For this reason 51Degrees.mobi is fully compatible with .NET version 2 and Visual Studio 2005 upwards.

The skills required to create mobile web pages are identical to those needed to create the existing web site. The documentation provided with our product explains through examples and downloadable code how simple extensions have been provided to ASP.NET components with which developers will already be familiar. We've tried hard to keep your business's move to mobile simple.

3.1.3 Fingers

Adult humans have similar sized fingers, and fingers are used for pretty much everything. From piano keys, to lift controls, to stone age tools, to car keys, to the old fashioned QWERTY keyboard, finger sized has been at the heart of design for a very long time.



The small size of the mobile device has thrown up a new design challenge that has yet to be solved successfully. How to enable humans to interact with the mobile device?

When only 10 digits were needed on mobile phones, fingertip sized was possible. This is particularly true of older and larger mobile phones. When letters were initially required for SMS messages number keys were adapted to support letters via multiple presses of the same key. T9 dictionaries were then introduced to speed up the entry of common words through prediction and eliminating invalid words.

Evolving from these early digit based phones we've seen touch screens, joysticks, track balls, slide out QWERTY keyboards, voice recognition, styluses and many other methods being deployed in production mobile devices. Unlike the desktop computer with its full size QWERTY keyboard and mouse, no single method has yet gained acceptance and all seem to be here to stay for some time to come.

Therefore the developer of a mobile web application has to consider two new factors:

1. How to make selecting items on screen easy.
2. How to optimise text entry.

3.1.3.1 Selecting Items

If selectable items on the screen are too small for a human finger to touch reliably when using a touch screen there's a reasonable chance the wrong item will be touched. The user may have wished to follow one link but has in fact selected another. This will infuriate the user and reduce the success of the application. Alternatively if there are so many selectable items on screen, a common occurrence with web sites designed for desktop browsers, then it may become impractical for the user to view them all when using a joystick or trackball.

51Degrees.mobi will automatically optimise the size of selectable items for the mobile device being used. Whether you're using hyperlinks, drop down lists or buttons they'll become easily selectable by human fingers.

If you want to have hundreds of selectable items on a single page then we're not going to stop you. But we'd seriously ask you to consider how you present your pages and suggest keeping the number of selectable things to a minimum. Very often a process that would involve a UI containing 5 steps in a traditional web browser will need to be altered to include 2 or 3 sub steps for each previous one to become optimised for the smaller screen of the mobile device.

3.1.3.2 Entering Text

Full size QWERTY keyboards are the easiest way to enter text into a computer. Mobile devices cannot utilise this input method and invariably the text input methods available are a compromise. That's why most people

don't write long emails on their mobile devices. It's important to keep the amount of text entry in a usable mobile web application to a minimum.

51Degrees.mobi has provided some tools to help.

Text box history will allow text that was previously entered to be selected again rather than re-typed irrespective of the capabilities of the phone. History can be shared across text boxes on different pages. For example; text boxes requesting search information can share the same history always enabling the user to select from the last 5 entries.

Common entries can be provided for selection based on the type of data being requested or the context of the application. For example; a person's full name often requires a title to be provided. Many people use the title Mr, Miss or Mrs. These 3 entries could be provided for easy selection thus avoiding the need to use the mobile device's text input method. See Figure 4 - Text box with selectable items.



Figure 4 - Text box with selectable items

Alternatively if the application requires the name of a location that could be near to you the 5 nearest locations could be made available for selection.

The important point is to reduce the barriers between the application and the user to improve usability and ultimately the success of the web site.

3.1.3.3 Conclusion

Different geographies and demographics will have a tendency towards different types of input method. In late 2010 middle class, 25 to 45 year old Europeans and North Americans are increasingly purchasing smart phones with high resolution displays and touch screen keyboards. India's middle classes are purchasing devices with physical keyboards and smaller screens, the smart phone being slightly too expensive to justify. When designing a mobile web application some thought needs to be given to the type of mobile device on which it is likely to be used so that the features available in the 51Degrees.mobi product can be used appropriately.

3.1.4 Performance



Web sites targeting desktop PCs connected to the internet via broadband connections have been able to provide a fast and reliable experience because a modern fixed line broadband connection provides so much fast data capacity. In our opinion many technologists and web designers no longer seriously consider the performance of the web sites they create because there are so few technical limitations.

When creating mobile web applications performance needs to be considered in a new context. The radio component of the data pipe represents an uncontrollable bottleneck. See Figure 5 - Mobile bottleneck.

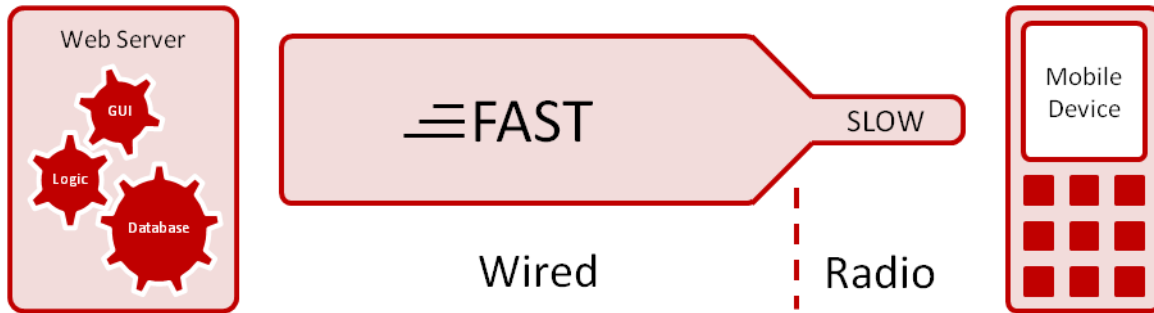


Figure 5 - Mobile bottleneck

Distance from the radio transmitter, atmospheric conditions, the radio spectrum in use and number of other concurrent mobile devices all play a major part in the available capacity and speed of the end to end connection between mobile device and web server. The same mobile web application used in the countryside in marginal coverage, or on a busy train will not perform as well compared to use at home via a WiFi connection with a broadband connection. Unless the mobile web application will only be used in a scenario where high capacity and speed can reasonably be assumed, such as WiFi connection in an office or home, the worst case scenario needs to be considered.

A second consideration depending on the target market should be the cost to the end user. Unlike the desktop internet where the web site owner and the customer pay a fixed price for their internet connection, mobile networks will often charge customers based on the amount of data consumed. See Figure 6 - Usage Charges.

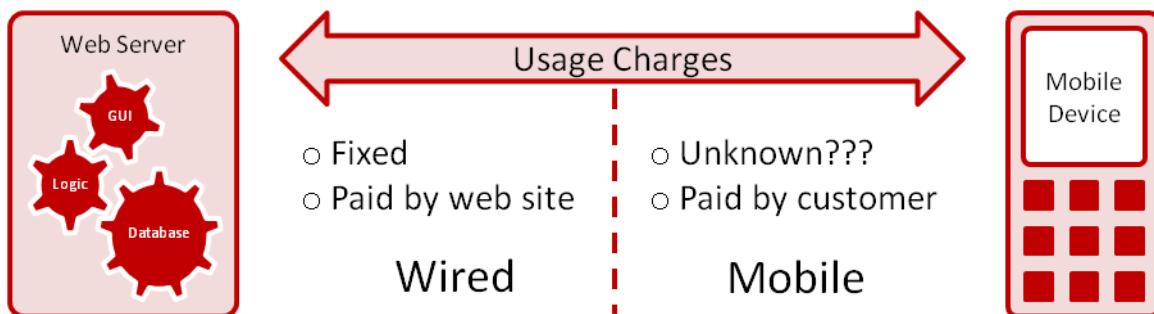


Figure 6 - Usage Charges

By reducing the amount of data sent between the web server and the mobile device the cost to customers can be reduced if they're charged based on the amount of data consumed. This may be less relevant if the mobile web application is targeting users of WiFi connected devices or will be used by known individuals with inclusive mobile data plans. However for any mobile web application targeting a broad mix of customers who may be charged based on the amount of data consumed this is an important consideration.

51Degrees.mobi contains many technical solutions to maximise the performance of mobile web applications, and reducing costs to customers, by reducing the amount of information that needs to travel between the web server and the mobile device. Our tools move the technology burden associated with common web features that can negatively impact performance and cost from the mobile device and the radio bottleneck to the web server without compromising the customer experience.

3.1.5 No Zoom

51Degrees.mobi actively disables zoom features on phones that allow zooming in their browsers such as the iPhone and Android devices. We've consciously made this decision because zoom functions have been developed out of necessity to enable mobile devices to be able to access web sites designed for big screen browsers. If you're designing a mobile web application specifically for mobile devices there should be no need to require users to change the zoom setting of the page. 51Degrees.mobi enables a perfectly sized page to be displayed on any mobile device eliminating the need for zoom.



3.2 Technical Design Concepts

Microsoft first released [ASP.NET in January 2002](#). It was designed to meet the needs of desktop web browsers and has proven to be an extremely flexible platform powering approximately 20% of the world's busiest million web sites.

ASP.NET originally contained some features to support mobile devices, notably XHTML and WAP Markup Language (WML). These features have become less relevant as modern mobile devices have started to use browsers containing features similar to desktop browsers. Microsoft deprecated mobile features in version 4.0 of .NET.

However some features designed into ASP.NET or web design in general are not mobile friendly. Examples include; sending and receiving cookies whether they're required or not, using images that are too large for the area in which they're being displayed, or persisting data using hidden fields in the web page. There are many more examples. All of these have the potential to seriously degrade the user experience.

Steps towards standardisation of technology have helped to solve some of the problems related to developing mobile web applications. Significantly WML no longer needs to be considered. However a number of important considerations remain:

1. Screen sizes and input methods vary widely between mobile devices.
2. Assumption cannot be made concerning available bandwidth or latency.
3. Mobile web applications typically have a different usage profile to web sites accessed via desktop web sites.
4. Mobile device manufacturers have not adopted a single standard.

We'll cover each of these four points as we explain the technical design concepts behind 51Degrees.mobi.

3.2.1 Wrapping ASP.NET

51Degrees.mobi wraps around ASP.NET to preserve all the inbuilt functionality whilst optimising performance of mobile web applications. See Figure 7 - 51Degrees.mobi wraps ASP.NET.

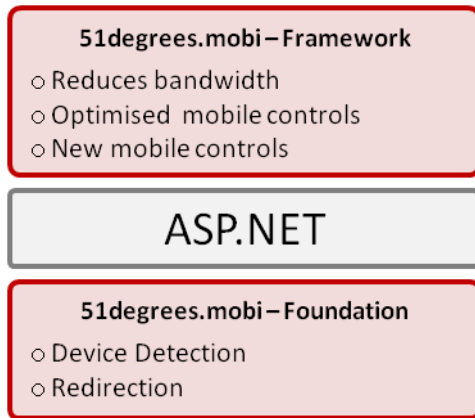


Figure 7 - 51Degrees.mobi wraps ASP.NET

The free and open source Foundation component fits under ASP.NET providing accurate device data to developers and providing low level functionality to redirect requests from different mobile devices to different pages. A mobile device can be directed to a page that is different from that used with traditional desktop web browsers. See 51Degrees.codeplex.com for more information.

The Framework component optimises the data sent and received from mobile devices reducing bandwidth, as well as enhancing standard ASP.NET controls and providing new ones for common functions.

Of most importance to any mobile web application is the need to minimise the amount of data that needs to be transferred between the mobile device and the server. See Figure 8 - Server to Mobile. ASP.NET is not always efficient in this regard. 51Degrees.mobi has introduced a number of enhancements to reduce data consumption.

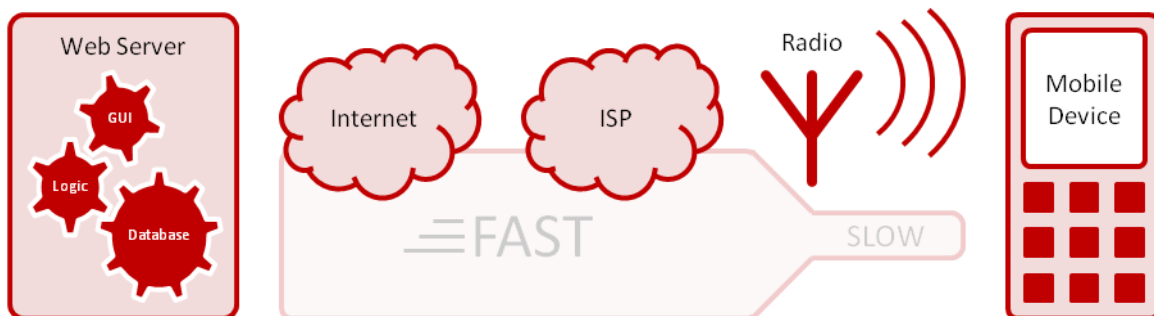


Figure 8 - Server to Mobile

This section covers the key technical design concepts used within the 51Degrees.mobi Framework to optimise ASP.NET for mobile.

3.2.2 Efficient Hyper Text Markup Language (HTML)

eXtensible Markup Language (XML), on which HTML is based, has been designed for human readability ahead of optimising data efficiency. The elements and attributes that it contains often include naming designed to

ensure the person creating it understands what is happening. For example; elements on a page contain a unique English identifier, styling is applied using English names such as border, margin or display. This is very useful when creating HTML but is not needed when the pages are used in a production environment as the underlying HTML will never be viewed directly by a human. 51Degrees.mobi contains features to minimise the amount of data that needs to travel through the mobile bottleneck shown in Figure 5. The three most important features are:

- Short element IDs of usually no more than 3 characters.
- Separating content and format.
- Stripping redundant data and compression.

An overview of each of these methods follows. If you are unfamiliar with Cascading Style Sheets (CSS) then we suggest you read [this](#) introduction before continuing.

3.2.2.1 Shorter IDs

A unique ID is assigned to any HTML element of significance to enable it to be identified within the overall HTML document. By convention in ASP.NET these are usually based on some meaningful English name. As a result they can become unnecessarily long.

ASP.NET will create HTML ID values based on the containing control and the ID the developer assigned to the control. The following is ASP.NET for an image control.

ASP.NET

```
<asp:Image id="ImageGlobe" runat="server" ImageUrl="~/Images/Globe256.gif" AlternateText="Globe"/>
```

The following HTML will be generated from the above ASP.NET.

HTML

```

```

In this instance the control ID is ImageGlobe. If this image control were contained within a user control and/or a master page then the ID produced will be prefixed with these containing control IDs to ensure it is unique across the entire page. As a result the ID appearing in the HTML could be very long, maybe 30 characters or more. If ASP.NET did not function in this way there is a possibility of two or more elements within the HTML sharing the same ID which would result in havoc during page processing and unpredictable results for the user.

Using 51Degrees.mobi all the IDs used in HTML are reduced from their English name to a 1 to 3 letter unique value significantly reducing the overall size of the page. The following HTML shows a 70% reduction in the number of characters needed for the same element id.

HTML

```

```

3.2.2.2 Separating Content & Format

ASP.NET enables controls to be formatted using two broad techniques. The first method uses attributes of the control such as Width, Height or BorderWidth to be specified. The second enables Cascading Style Sheets (CSS) styles to be applied directly to the control to provide format information. The following is ASP.NET for an image control with a 2 pixel wide solid border and an image that is 50% the width of the screen.

```
<asp:Image id="ImageGlobe" runat="server" ImageUrl="~/Images/Globe256.gif"
AlternateText="Globe" Width="50%" BorderStyle="Solid" BorderWidth="2px"/>
```

The above ASP.NET will produce the following HTML.

```

```

An alternative method that will produce the same HTML would be to specify the CSS style attribute directly instead of providing the attributes Width, BorderStyle and BorderWidth using the following ASP.NET.

```
<asp:Image id="ImageGlobe" runat="server" ImageUrl="~/Images/Globe256.gif"
AlternateText="Globe" style="border-width:2px;border-
style:Solid;width:50%;"/>
```

Either method will produce the same result.

This approach to formatting controls is very effective because the developer can see the format information alongside the control content. The problem with providing format information in this way is that the HTML produced contains repetitive information that has to be sent from the web server to the mobile device every time the page is produced.

51Degrees.mobi will automatically remove format information from the HTML and provide it via a separate document. This approach improves performance on subsequent times the page is accessed because the style information is already held on the mobile device and does not need to be retrieved again.

Figure 9 shows how the HTML document is split during the 1st request and the mobile device will 1st retrieve the content and then the format.

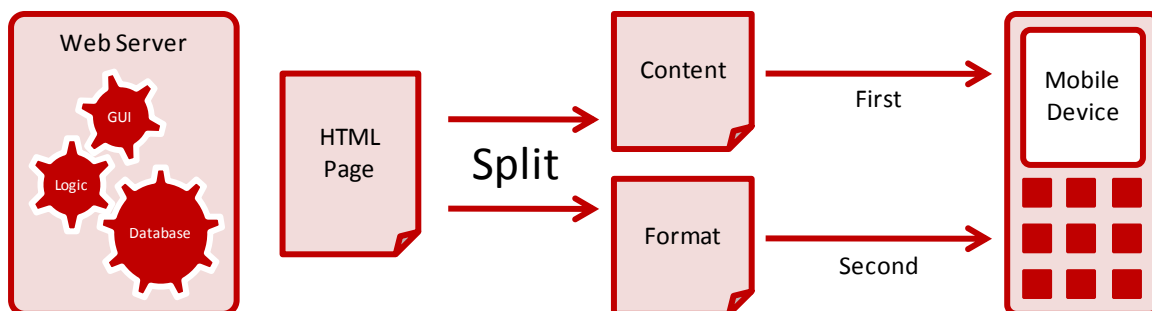


Figure 9 – 1st Request - Separating Content and Format

Figure 10 shows a subsequent request for the same page type, but with different content. The page is split in the same way as the 1st request. However the mobile device already knows about the format information and does not retrieve it a subsequent time. This approach minimises the amount of data sent between the web server and mobile device over multiple requests for the same page. If the web site contains a small number of page types accessed frequently this approach can significantly improve performance.

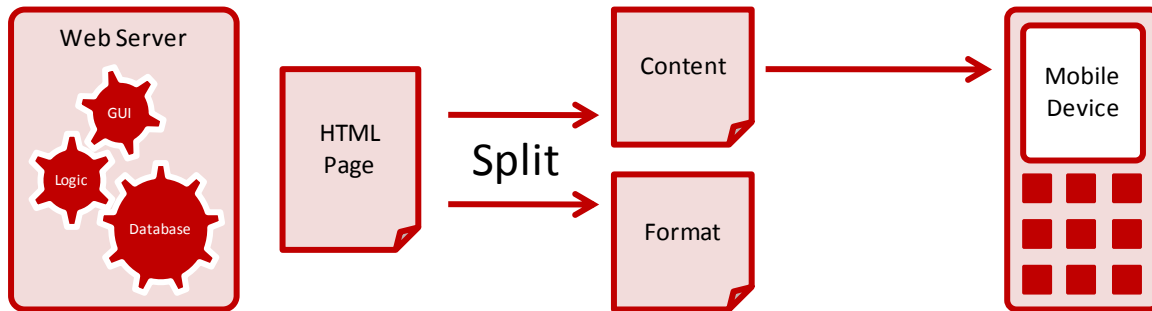


Figure 10 - Subsequent Requests - Separating Content and Format

The web site designer can also use custom designed CSS documents rather than providing format information within the ASP.NET. However due to the sheer number of different mobile devices and browsers such an approach should only be considered if the type of mobile devices being used can be predicated confidently, or the consequences of the device applying the CSS style incorrectly are minimal. 51Degrees.mobi recommends using the format attributes of controls to apply style information rather than CSS.

3.2.2.3 Stripping Redundant Data and Compression

Before sending documents to the mobile device 51Degrees.mobi will check the document for any data that will not be used by the Mobile Device. Examples of such information include; comments added by the web designer to aid understanding and white space included to make it easier for a human to distinguish the different parts of the document.

Additionally if the mobile device supports the ability to uncompress data the document will be compressed by the web server. Compression occurs within the 51Degrees.mobi framework irrespective of the configuration of other components within IIS.

3.2.3 Cookies & Mobile Profiles

Many web sites persist data within the web browser through the use of cookies. Cookies are small files held within the web browser that can be retrieved by the web server. If present they're sent to the web server by the browser every time a web page is requested. Whilst they're often small they can have a noticeable impact on the amount of data consumed, especially when they're not actively needed by the request being processed. 51Degrees.mobi does not prevent the use of cookies. We provide an alternative that should be considered in almost any situation cookies could be used.

Mobile Profiles are stored in a database within the web server environment. Each time a new mobile device is seen by the web server a new mobile profile will be created. The mobile profile can be used to store information that should only be associated with a specific mobile device. As the information is held within the server environment there is no need to use browser based cookies.

Mobile profiles are used by 51Degrees.mobi controls to persist data about previous requests from the device. For example; when a drop down list is used the selected value can be stored in the mobile profile. The next time the same drop down list is displayed the previously selected value can become the default. In this way the user will not need to reselect the drop down list value if they don't wish to change it. Mobile Profiles should be used to store any information that in a non mobile environment would be stored in cookies.

Mobile profiles also provide access to information about the environment in which the mobile web application is operating. A key piece of information exposed via mobile profiles is the currently available bandwidth

between the web server and the mobile device. This information could be used to tailor the experience, perhaps removing the display of some images if the bandwidth is particularly low.

Information stored in a mobile profile is available outside the standard browser based interaction with a mobile device. Mobile profiles can be used within a text message based application to persist information associated with previous text messages. A hybrid text message and mobile web application could be created that shares data between the two mediums.

[Moore's law](#) predicts; "The number of transistors that can be placed inexpensively on an integrated circuit has doubled approximately every two years". Changing some of the underlying mechanisms used in ASP.NET is only possible because of the exponential growth in computing power since ASP.NET was originally designed and released. See Figure 11.

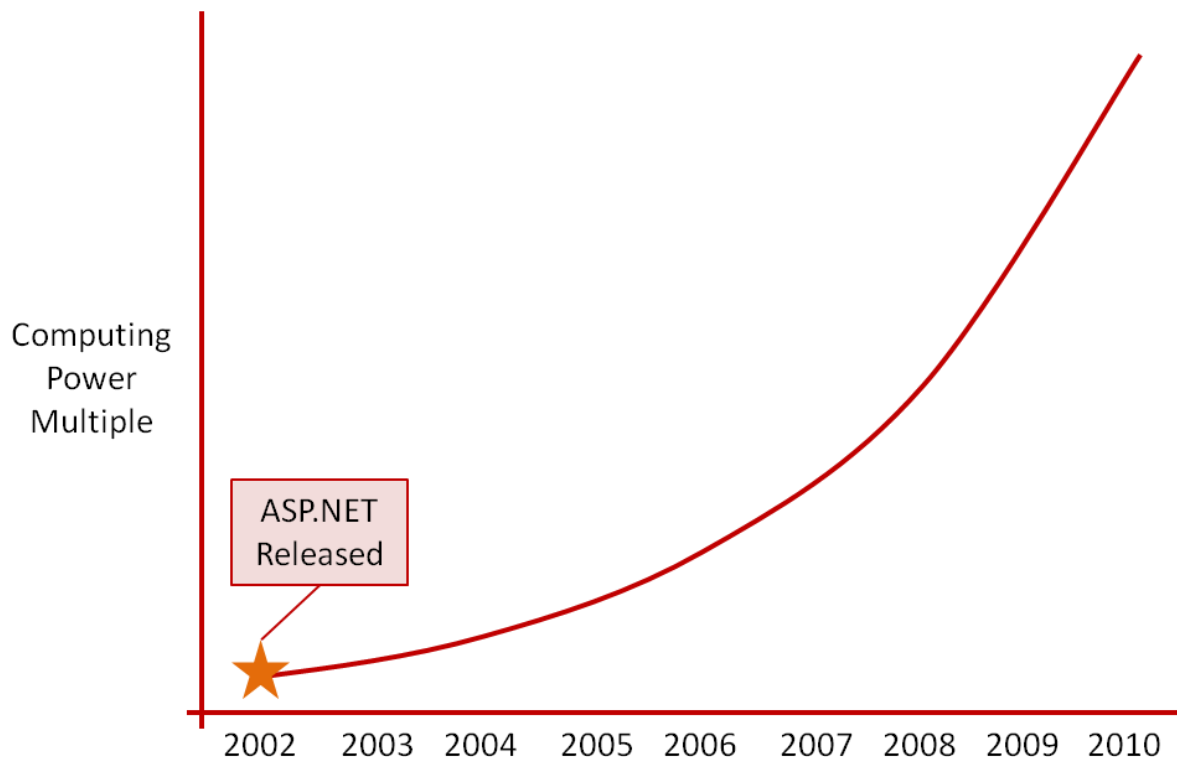


Figure 11 - Moore's law since ASP.NET launched in 2002

Inexpensive SQL Server databases now perform so rapidly, core ASP.NET tasks can be moved to the database without a reduction in site performance. Mobile web applications will perform faster because the radio component of the data path between the mobile device and the web server can often be the bottleneck slowing performance.

Mobile profiles place the burden of data persistence on the web server environment exclusively. For large web sites this may require investment in database infrastructure. However they reduce the burden on the mobile device and radio component of the end to end service. We believe the trade off will be worth it.

3.2.4 Images - Size Matters

Many web designers will design a web site with a fixed pixel width in mind. Often such web sites will be centred within the browser and a background image or colour will extend to each edge of the display. When creating web sites for the small screen of the mobile device there is no spare screen space to indulge in such a

luxurious approach. The page needs to fill the screen and relevant content provided at the top of the page. Designers of mobile web applications should avoid specifying width and height in pixels. Instead percentages and point sizes should be used.

This approach creates some significant issues when displaying images. Images are data hungry and are created with a specific pixel width in mind. If a web design requires an image to fill the width of the screen how many pixels wide should it be? See Figure 12 - Example Picture with 100% Screen Width.

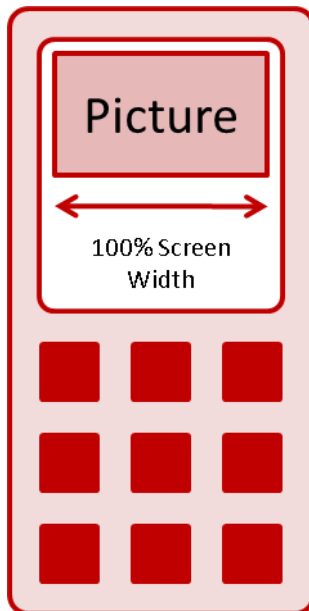


Figure 12 - Example Picture with 100% Screen Width

An approach that could be taken using ASP.NET without 51Degrees.mobi would be to provide an image the size of the largest screen width of device currently available. Assuming this was 640 pixels, a mobile device with a 320 pixel wide screen would download the full 640 pixel wide image and then reduce it to fit the screen width. This approach will result in nearly twice the amount of data needing to be download than is actually required.

With 51Degrees.mobi enabled an image will be provided by the web server that is precisely sized for the device requesting it. This approach significantly improves performance, user experience and reduces cost to the customer.

3.3 Target Audience

51Degrees.mobi is a product designed to be used by people familiar with the ASP.NET technologies provided by Microsoft and the documentation is written based on the assumption the reader will have a basic understanding of ASP.NET. If you've not used ASP.NET previously we recommend you familiarise yourself with the fundamentals using a beginners' guide. The following may be worth considering:

- [Beginning ASP.NET 2.0 with C# \(Wrox\)](#)
- [ASP.NET for Dummies \(For Dummies\)](#)

Others are available. 51Degrees.mobi does not endorse any of these books.