

Browser testing guidelines

Version:	1.0
Date created:	13/11/2008
Policy official:	Adam Bailin
Date last updated:	09/01/2009
Date issued:	January 2009
Lead official:	David Pullinger
Guidance number:	TG117

Purpose

This guidance is for all website managers, web developers and web testers delivering public sector websites. It is designed to ensure that your website works on as many browsers and devices as possible. It also provides guidance on the browsers and operating systems with which to test your website.

Table of contents

Introduction	2
Aims and objectives	2
Who is this guidance for?	2
Scope	2
What you need to know	3
Which browsers are accessing my site?	3
How often should I update my testing matrix?	3
Do I need to publish the testing matrix?	3
Do I need to test my website with every browser?	4
Code to standards.....	5
How to develop a testing matrix	6
Test with popular browsers	6
Test with current versions	6
Include Windows, Mac and Linux.....	6
Test with mobile devices	7
Advice on testing	8
Testing tools.....	8
Example testing matrix.....	9
Acknowledgements	10

Introduction

Aims and objectives

The Government aims to provide a consistent high quality experience for users across all of its online services. There are a large number of different browsers on the market and to test your website with all of them would be impractical and inefficient. However, if you build your website to comply fully with standards, this effort will be significantly reduced and you may only need to test your website with a few browsers.

The objective of this guidance is to ensure that you website works on as many platforms and devices as possible. An effective way to do this is to:

- i. code to standards;
- ii. test with browsers.

If you develop standards-compliant code (eg (X)HTML, CSS, JavaScript etc) you can be confident that your website will work on a wide variety of platforms and devices. If you then test with a few popular browsers you can be confident that the website works for as many users as possible. This guidance aims to help you decide which browsers to test by building a matrix of different browser and software platforms – a *browser testing matrix*.

Who is this guidance for?

This guidance is for public sector website managers, developers and testers:

Website managers

- [How to find out which browsers are accessing your site](#)
- [How often your testing matrix should be reviewed](#)
- [An example browser support statement](#)

Web developers and testers

- [Developing standards-compliant websites](#)
- [How to determine which browsers to test with](#)
- [Help with testing](#)

Scope

In scope:

- Web standards
- How to decide which browsers to test with
- When you should review your testing matrix

Out of scope:

- Browser security testing
- Accessibility and assistive technology testing

What you need to know

Which browsers are accessing my site?

1. There are several websites which provide global statistics for browser usage, however these figures may not be representative of the users of public services or your website in particular. The [BBC's Browser Support Standards](#)¹ are a good starting point.
2. Your testing matrix must include the most popular browsers accessing your website.
3. Information on browser usage is available from your web server log files or from web analytics servers.
4. Different target audiences may have different browser preferences. Therefore, your testing matrix should also take into account target audience demographics.
5. Consider including browsers in your testing matrix even if they do not show up in the logs. This might be because the current site design doesn't work on a particular browser, leading to a decline in its overall usage.

How often should I update my testing matrix?

6. Your browser testing matrix should be updated at every major website re-development to ensure that the latest versions are catered for.
7. Reviewing the matrix should also be considered if a market leader (i.e. the most popular supplier for either operating system or browser) plans a major version upgrade. These are typically an x.0 rather than a 0.x release.
8. As a minimum, your testing matrix must be updated every two years.

Do I need to publish the testing matrix?

9. You should publish a list of the devices you have tested your website with, including browsers, mobile devices and assistive technologies.
10. The list should be directly linked from the home page, for example as part of your 'Help' or 'Accessibility' section.

¹ BBC Browser Support Standards v3.3
http://www.bbc.co.uk/guidelines/futuremedia/technical/browser_support.shtml#support_table

11. Published information must be updated whenever the testing matrix itself is updated.
12. Avoid using statements such as, 'This page is best viewed with Browser X'.
13. The following is an example browser support policy statement that may be used on your website:

[Website name] has been developed to comply with the following web standards and then tested with a wide range of devices:

[List of devices used for testing]

If you are experiencing problems with accessing the site, please contact [details here].

The following browsers should also provide access to all of the content and functionality on the site:

[List of browsers not used for testing but assumed to work]

Do I need to test my website with every browser?

14. Public sector websites have a responsibility to be inclusive and not to exclude groups of users. This means that your browser testing matrix should be as inclusive as possible. However, this needs to be balanced against increased testing times and costs.
15. If you code your website to standards then testing requirements can be significantly reduced. However, your testing matrix should include the three main operating systems: Windows, Mac and Linux.
16. There may be specific browsers that you choose not to test because they are either old or unpopular. The following guidance has been devised to help you decide the most appropriate browsers to test.

Code to standards

17. Coding a site to web standards should ensure that any browser that supports web standards will render and behave as intended. Therefore your browser testing matrix must include browsers that support web standards.
18. You should follow a *progressive enhancement* approach to developing websites to ensure that content is accessible to the widest possible number of browsers. This approach to web development works by building your site in layers:
 - i. Code semantic, standards-compliant (X)HTML
 - ii. Add a presentation layer using CSS
 - iii. Add rich user interactions with JavaScript
19. The [guidance on minimum technical standards](#)² highlights the importance of developing websites to standards to ensure browser accessibility. The following guidance builds on that advice.
20. You must choose a DTD (Document Type Definition) for your (X)HTML content. It is recommended that you use a *strict* DTD, however for legacy content or sites that import comments (e.g. blogs) it may be better to use *transitional*.
21. All (X)HTML content must validate with respect to your chosen DTD.
22. You must use valid CSS for the presentational layer of your website including layout and styling. (X)HTML tables should only be used for presenting tables of data.
23. Code used for adding richness to the user interface (e.g. JavaScript, ActionScript) must be ECMAScript-compliant.
24. However, browsers have varying support for web standards, and [popularity is a major factor](#) as well.

² Minimum technical standards (TG109) <http://www.coi.gov.uk/guidance.php?page=176>

How to develop a testing matrix

Test with popular browsers

25. Testing with browsers and operating systems used by 2% or more of your users should lead to a reasonable level of browser support. This does not imply that browsers used by less than 2% of your users are unsupported. However, the 2% figure is a useful guide to where to spend time testing and fixing issues.
26. Browsers used by 2% or more of your users must be tested, and any issues resolved.
27. Browsers and operating systems used by less than 2% of your users are not required to be tested. Minority browsers that support standards are assumed to work.
28. Web analytics for browser usage are usually based on numbers of unique users, visits or page impressions. To ensure equal priority is given to each user, determining which browsers appear in your testing matrix should be based on unique users.
29. Unique users are defined according to the industry standard definition from the [Joint Industry Committee for Web Standards](#)³ (JICWEBS).

Test with current versions

30. Older browser versions need not be tested on newer operating systems (i.e. those shipped with a later release date).
31. You should test fully released browsers and operating systems in favour of pre-released betas, where possible.
32. You should test with the latest service pack or major maintenance release (or point release) of an operating system. Minor maintenance releases need not be tested separately.

Include Windows, Mac and Linux

33. The Linux and Mac operating systems should be included in your testing matrix even if they account for less than 2% of your unique users.

³ Joint Industry Committee for Web Standards <http://www.jicwebs.org/standards.php>

34. Operating systems that are not supported by the supplier from a security perspective (e.g. Windows 98 and Windows NT) are not required to be tested.

Test with mobile devices

35. Mobile is a rapidly growing channel with huge market penetration and therefore you should test your website with mobile devices. However, there are a large number of varying devices and to test your website with all of them would be impractical.
36. To get an idea of how your website works on different devices, you may wish to use software emulators that mimic mobile devices and allow you test without acquiring a large number of devices. There are three main options:
 - Desktop browsers – add-ons or features of desktop browsers that simulate browsing with a small screen device.
 - Web-based emulators – websites that simulate mobile devices; these have greater accuracy than desktop browser simulators.
 - Desktop emulators – desktop software that simulates mobile; often provided with software development kits, these are probably the best for testing specific devices.
37. For communications targeted specifically at the mobile channel, you should consider your target audience and test with the most popular makes and models for your target audience (e.g. 5-10 varying devices).

Advice on testing

38. When testing with different browsers, it may be useful to consider the following aspects of your website:
- **content** – is the message being communicated by the page delivered to the user?
 - **functionality** – can users do everything they need to?
 - **display** – does the page look as it should do in terms of style and layout?
39. You should check that the content, functionality and display all work as intended. However, there may be minor differences in the way that the website is displayed. The intent is not that it should be pixel perfect across browsers, but that a user of a particular browser does not notice anything appears wrong.
40. When testing website functionality, you may wish to consider different types of interaction, for example:
- Navigation
 - Form filling
 - Browser back button
 - Logging in to secure areas
 - Sessions or cookies
 - Bookmarking (i.e. when content is bookmarked, does it return to the same content exactly without requiring any further interaction?)
41. Some content can deliver an engaging user experience that may be unavailable to users with scripting disabled or without the correct browser plug-in. For example, corporate users may be excluded due to their organisation's IT policy. Therefore, you should also test your website to make sure that it works with scripting and plug-ins turned off.
42. Some users will be unable to use pointing devices so you should verify that the site works using a keyboard only.

Testing tools

43. The following list is a selection of tools that may help with testing your website:
- [W3C Markup Validation Service](http://validator.w3.org/)⁴ – quick way to test (X)HTML pages for valid markup.

⁴ <http://validator.w3.org/>

- [Total Validator](#) – web testing tool that includes screenshots of pages in different browsers.
- [W3C CSS Validation Service](#)⁵ – quick way to test the validity of CSS style sheets.
- [JSLint](#) – JavaScript verifier.
- [Total Validator](#) – web testing tool that includes screenshots of pages in different browsers.
- [Firefox Web Developer toolbar](#)⁶ – useful tool for building standards-compliant websites.
- [VMware](#) – virtualisation software, allows you to run several virtual operating systems on one machine.
- [Litmus](#) – web-based testing tool; an alternative to using hardware.

Example testing matrix

44. The following is an example testing matrix. It is based on the [BBC's testing recommendations](#)⁷ at the time of writing. You may use this matrix unless your target audience browser usage deviates significantly from the recommended set.

Browser	Firefox	IE	Safari	Chrome	Opera
Platform	All	Windows	Mac	Windows	All
Must test	3.0.x	6, 7	3.x	0.2.x	9.x
Should test	2.0.x	5.5	2.x		8

⁵ <http://jigsaw.w3.org/css-validator/>

⁶ <http://chrispederick.com/work/web-developer/>

⁷ BBC Browser Support Standards v3.3

http://www.bbc.co.uk/guidelines/futuremedia/technical/browser_support.shtml#support_table

Acknowledgements

The Central Office of Information would like to thank the following organisations for their contribution to the development of this guidance:

IBM Ltd.

BBC

British Computer Society

Department for Innovation Universities & Skills

Her Majesty's Revenue & Customs

Nomensa Ltd.

The Open Source Consortium

Opera Ltd.

Reading Room Ltd.

The Web Standards Project