



How to Commission & Design Accessible Websites

For Voluntary and Community Organisations

Produced by



Copyright Statement

All rights reserved. This publication is protected by copyright law. Material may be used or copied for personal use or for internal use within your organisation only. If you make unauthorised copies from these pages, the publisher could take you and your employer to court and claim legal damages.

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act, 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers. Every effort has been made to make this publication as complete and accurate as possible, but no warranty of fitness is implied. AbilityNet does not represent or warrant that the information contained in this book is accurate, complete or up to date.

Neither AbilityNet nor the ICT Hub shall have liability or responsibility to any persons or entity with respect to loss or damages arising from the information contained in this book.

Copyright © 2006 AbilityNet

Contents

Introduction	4
Part one: An overview of web accessibility for the VCS	6
What is web accessibility?	6
What are the benefits?	7
Facts and figures	8
An introduction to accessibility guidelines	9
Who does it affect?	10
Future trends in web accessibility	15
Part two: For commissioners	16
Introduction	16
What the law says about web accessibility	16
Planning for an accessible website	18
Web Accessibility Policy	20
Working with designers and developers	21
Post launch – how to keep your site accessible	22
Part three: For editors: Writing for the web	23
Language	23
Pictures and symbols to help understanding	24
Writing alternative text	26
Hypertext links	27
Headings	28
Editors checklist	28
Part four: For developers and designers	29
Web accessibility in context	29
Web accessibility level 1 issues	30
Key Level 2 accessibility issues	38
Word, PowerPoint and PDFs	46
Part five: Testing the accessibility of your website	48
Testing overview	48
Disabled user testing	52
Testing for specific accessibility issues	54
Testing checklist	58
Part six: Resources	60
Glossary	60
Useful websites	62
W3C Web Content Accessibility Guidelines Checklist levels 1 and 2	63
Helpful organisations	66

Introduction

This guide is designed for all voluntary and community organisations to explain why web accessibility is vital to support your work with beneficiaries, and how easy it is to make simple practical improvements to the accessibility of your website.

The booklet is part of a web accessibility pack produced by AbilityNet, the UK's leading authority on accessible IT, for the ICT Hub.

The first part of the booklet gives an overview of web accessibility. It explains what web accessibility is, the benefits for your organisation and the wide range of people it helps and explains some of the key future developments in this area.

Section two is specifically for anyone who commissions or manages a website. It explains what the law says about accessibility, gives guidance on planning for an accessible website and talks you through a number of key considerations that are important for making your website accessible. These include creating an accessibility policy and working with designers and developers.

Part three goes into the detail of writing content for the web - a section specifically for anyone who creates content for your website. Part four details all the main accessibility issues that impact on the build and design of your site. Part five takes you through how to test your site for accessibility using free tools and techniques.

Part six contains useful resources such as a glossary of key accessibility terms, useful websites to take your knowledge of accessibility further and the World Wide Web Consortium's (W3C) Web Content Accessibility Guidelines (WCAG) checklist levels 1 and 2.

The accompanying CD contains web links, information about accessibility tools, free web templates and video demonstrations of accessibility issues to complement this guide.

What is Accessible Technology?

Accessibility is about making changes or adjustments to computers so they are easier for the user to access. All organisations need to ensure that their computers and websites can be made accessible to those who have difficulty reading the screen, problems using the keyboard and mouse or reading and writing difficulties. In most cases this can be achieved through making minor low cost changes to system settings or websites.

Who are AbilityNet?

AbilityNet is a national charity and the leading authority on disability and computing. AbilityNet provide Assessment, Training and Consultancy in accessible computing for the commercial, public, not-for-profit sector and for individuals.

Our assessment service recommends individually tailored IT solutions for people who find some aspect of using a computer difficult. We have a wealth of experience in training and consultancy to organisations of all sizes tailored to suit their requirements.

AbilityNet has a strong reputation for working with community and voluntary organisations as well as those involved in adult training. Web accessibility consultancy is one of our key services; we work with a wide range of organisations such as Citizens Advice, BBC and recently with Ufi learndirect on the development of its Myguide website. We also publish quarterly a state of the eNation report which takes a snapshot review of web accessibility in different sectors, the report in April 2006 looked at the voluntary and community sector.

Introduction

What is the Information and Communication Technologies (ICT) Hub?

We are a group of voluntary sector organisations who have come together to plan and deliver a co-ordinated framework of ICT guidance, good practice, advice and support for voluntary and community organisations, accessible at a local level.

Our aim is to improve voluntary and community sector ICT infrastructure so that voluntary and community organisations are enabled to achieve their missions more efficiently and effectively through the better use of ICT.

Did you know?

60% of internet users would see their user experience enhanced by accessibility features, regardless of impairment

Forrester Research 200

Part one:

An overview of web accessibility for the VCS

What is web accessibility?

Web accessibility is about designing sites so as many people as possible can access and interact with them effectively and easily, independent of who they are or how they access the net.

This particularly benefits disabled and older people, many of whom see the web as an empowering medium giving them access to information and services that they either would not have access to in person or would have to rely on other people to provide.

Adaptive Technology

Disabled people will often use something called adaptive or assistive technology to access the web. For example blind web users will use software called a screen reader which reads aloud the content of a web page; Jaws, Hal and Window-Eyes are popular examples of screen readers.

Adaptive software relies on web pages being designed accessibly to function properly, if they aren't then disabled web users will frequently have a hard time understanding what is on a web page.

Changing Browser options

Many disabled people particularly those with mild visual impairments or dyslexia don't need to use adaptive technology to access the web, instead they rely on being able to change how a web page looks by increasing the text font size or by applying different colour schemes or background colours, for example people with dyslexia often find black text on a beige background the easiest to read.

Depending on how a website has been designed making these changes can either be very easy and straightforward if it is accessible, or it can be a complex drawn out process if a site has not been designed with accessibility in mind. Later in this guide we will talk in more detail about the types of disabled web users who have problems with inaccessible websites.

Why is web accessibility an issue?

- Lack of standards in the early days of the web.
- Ignorance of the needs of disabled web users.
- Development tools were very poor at creating accessible websites.
- Limited advice and support available.

What are the benefits?

Web accessibility is not just about meeting legal obligations; there are also numerous benefits, which include:

- **Reach a wider audience**

Accessibility makes your site easier to use not just for disabled people but also for 'silver' surfers, people whose first language is not English and other disadvantaged groups in society.

- **Makes your site more useable for everyone**

There is a strong relationship between usability and accessibility – research (April 04) in the UK by the Disability Rights Commission found a 35 % increase in usability for accessible sites in comparison to non accessible sites. So for example it will help more people make full use of your site facilities such as on-line donations.

- **Reduces site maintenance - cut costs**

Accessible sites are more efficiently coded, saving bandwidth and tend to be easier to maintain because they are more structured using a technique of separating the page code from the way it looks by using a technology called Cascading Style Sheets (CSS).

- **Helps with search engine rankings**

Coding your pages for accessibility has the added benefit of making them easier to 'read' for search engines like Google, increasing your visibility on the web.

- **Platform independence**

Accessible websites also work well with a range of devices such as PDAs, WebTV and also Mac and Linux PCs, and helps future proof your website.

- **Browser independence**

Designing your sites accessibly also means that it will work with a wider range of browsers and be backwards compatible with older ones.

- **Social responsibility and reputation**

Not only is it a good thing to do, but it will help you as a voluntary or community organisation make a real positive difference to disabled people in the community you serve, they will appreciate your commitment to making your site accessible to them.

Facts and figures

It is important to understand the scope and scale of inaccessible websites, below is a sample of statistics:

1. In 2004 an investigation into website accessibility for the Disability Rights Commission (DRC) found **81%** of **1000** websites tested across the public, private and voluntary sector in the UK did not meet minimum accessibility standards.
2. A conservative estimate is that between **1.3** and **3.2 million** disabled people in the UK have problems with inaccessible websites.

These figures do not take into account:

- People with Dyslexia (approximately 6 million).
- People with literacy issues (17 million).
- People whose first language is not English.
- Older population – silver surfers.

(Sources: DWP, AbilityNet)

3. **1 in 3** Britons are aged **50** or more, and they are the fastest growing segment of the internet population; by the year 2010, **40 %** of the UK population will be over **45** – the age at which the incidence of disability begins to increase significantly. (Source: EFD)
4. Approximately **32 %** of people using accessibility options or assistive technology do so for ease of use, comfort and convenience. They have no physical difficulty or impairment. (Source: Forrester Research 2004)
5. **8 %** of the United States population, (over 20 million people), have visual, learning, cognitive, auditory or mobility impairments severe enough to affect their ability to access the Web. (Source: Judy Brewer ,W3C)

An introduction to accessibility guidelines

There are a number of accessibility standards in circulation – ‘Bobby’, RNIB’s (Royal National Institute of the Blind) ‘See it Right’ and Section 508 (USA) to name a few.

However they all stem from the same set of guidelines produced by the World Wide Web Consortium (W3C) called the Web Content Accessibility Guidelines (WCAG).

The guidelines were created in 1999 to explain how to make websites accessible to people with disabilities. They are prioritised into three levels:

Priority Level 1 - ‘Must’ or level ‘A’

Minimum: failure to meet this basic level means many disabled people will not be able to access or interact with your website.

Priority Level 2 - ‘Should’, or level ‘Double-A’

Good practice: This level removes many of the barriers to accessing your site for a range of disabilities.

Priority Level 3 - ‘Ought’ or level ‘Triple-A’

Beyond best practice: This level removes all barriers to accessing your site.

Priority level 2 is the standard that many organisations are aspiring to reach over the next 1-2 years.

It is not safe to assume priority level 1 is the minimum for legal compliance in the UK, under the Disability Discrimination Act (DDA). Until there is a court case in the UK it is unclear what the web accessibility standard should be that meets ‘DDA Compliance’.

Version 2 of the Web Content Accessibility Guidelines

Version 2 of the guidelines is out in draft and is likely to come into force by the end of 2006. The principles are the same as version 1 but the emphasis has shifted to make the guidelines more technology independent. Version 1 focuses on html and Cascading Style Sheets (CSS), Version 2 has been expanded to cover other technologies such as PDF and Flash.

Who does it affect?

This section explains the accessibility issues of people with a range of impairments including physical, sensory and cognitive. It also highlights key practical issues with case studies to help you better appreciate the frustrations many disabled people experience with inaccessible websites .

Before we talk about specific disabilities, it is also important to understand that web accessibility is not just for 'disabled people' but also important for:

- People who use PDAs such as pocket PCs and Palms to access the web.
- People who have temporary impairments such as a broken arm or wrist.
- People behind secure firewalls that can block JavaScript for security reasons.
- Silver surfers over 50 who might have minor vision, mobility or hearing problems.
- People who access the internet away from the traditional desk environment for example on the road, via public internet kiosks or while doing other tasks that require the use of their hands.
- Any organisation which wants to maximise its profile in the community and attract disabled service users.
- Organisations that want to use their website to share information with the widest range of staff and volunteers.

People with specific disabilities that have problems with inaccessible websites can be broadly categorised into four key groupings:

- **Vision:** including no vision, colour blindness and tunnel vision.
- **Hearing:** both total Deafness and hard of hearing.
- **Mobility problems:** such as a physical difficulty using hands and arms, discomfort, impaired motor skills and fatigue.
- **Cognitive, Mental and Learning Disabilities:** such as dyslexia, attention deficit disorder and short term memory loss.

Adaptive Technology

Adaptive (or assistive) technology enables disabled people to access computer systems that would be inaccessible by conventional means.

Examples include:

- **Screen magnifiers** which help people with low to moderate vision impairments.
- **Voice recognition systems** such as Nuance's Dragon Dictate often used by people with mobility problems.
- **Alternative input devices** such as eye and head operated pointing devices and sip-and-suck systems controlled by breathing or a single switch device.
- **On-screen keyboards** are used by people who are unable to use a standard keyboard such as switch users. A switch is often a large button that can be pressed by any part of the body. It is particularly useful for anyone who has restricted movement but can reliably move one part of their body such as their head or hand to press the switch.

Case Study: David

“I am paralysed from the neck down and ventilated; I use a headset and on-screen keyboard to access my computer. Due to my severe disability the computer and especially the Internet is my primary communication link.

I use the web to access nearly all of my news and sport information as I am unable to hold and flick through a newspaper.


It also enables me to actively participate in my local community through web and email, working with a number of local charities and community organisations.”

Vision

As a very visual medium, the web presents unique problems to the millions who have low, restricted or no vision.

There are four broad groups of people with visual impairments that have different approaches to viewing web pages:

Colour blindness – there are at least three types but red/green impairment is the most common, where people find it hard to distinguish between the two colours. It affects approximately 5 % of men and 1 % of women. It is particularly important to test for colour blindness issues when creating a colour scheme for a website. You can test pages online at vischeck.com or download a Photoshop plug-in from the same site.

 See **Video 1: Colour blindness** on the Video demos section of the Resources CD for a practical demonstration.

Mild visual impairment - This group normally will want to enlarge the text size on web pages, and use their own colour scheme such as high contrast, for example white text on a black background. In Windows you can find the high contrast option under ‘Accessibility Options’ in the Control Panel.

 See **Video 2: Changing text size and background colour** on the Video demos section of the Resources CD for a practical demonstration.


Moderate visual impairment - people with a moderate visual impairment use assistive technology called screen magnifiers to view web pages. This can zoom the size of the page up to x32 times its normal magnification but many people use it as x4 or x5 magnification. A basic magnifier is built into Windows 2000/XP, you can find it under Accessories > Accessibility > Magnifier.

 See **Video 3 Magnification** on the Video demos section of the Resources CD for a practical demonstration.

Severe visual impairment - People with no or little residual vision need to use a more radical access technology to read web pages known as screen readers, they read out in synthesised speech the content of a web page. Screen readers are the most demanding adaptive technology for accessibility because they look at the code underneath the visual design to make sense of the web page.

Jaws, Window-Eyes, Hal and IBM's HomePage Reader are common screen readers. Also the new version of Mac OS 10.4 for the first time has an integral screen reader built in called VoiceOver. For more details on the Mac screen reader see:

<http://www.apple.com/accessibility/>

 See **Video 4: Screen reader - images** on the Video demos section of the Resources CD for a practical demonstration of a screen reader in action.

Case Study: Robin

Robin despite limited vision uses a computer very effectively, relying on speech output (screen reader) to access the full range of mainstream software including e-mail and the Internet.

He says:

“Technology can immensely increase the life choices of anyone with a disability. The computer is far more flexible and inclusive than we appreciate; the wealth of hardware and software available means that everyone is able to work and play to the full.”

“I find that the Internet is an increasingly powerful resource of information and services. Good practice in web design is important to enable everyone easy access to what the net has to offer, but vital for those, like myself, who use alternative methods of access.”

Robin works in a very demanding job as AbilityNet's Web Consultancy manager, responsible for a team of seven, promoting the benefits of web accessibility to the public, private and voluntary and community sectors.

Hearing

Hearing impaired people have particular problems with inaccessible multimedia, including video and audio clips on the web. This is a major issue for media sites that deal with streaming media.

Guidelines developed by the World Wide Web's Consortium's Web Access Initiative (WAI) recommend that web creators include text transcripts of any multimedia on a web page.

Additionally for those whose first language is BSL (British Sign Language) there are words in English that do not exist in their vocabulary. For example research by the BBC found terms such as 'marinade' in their recipe section were not understood by BSL users, and they realised they needed to provide a glossary for such cookery terms.

Also many Deaf or hard of hearing people's preferred method of electronic communication is via a textphone or Minicom. Your organisation does not have to own one you can instead use the TypeTalk service set up by RNID (Royal National Institute for the Deaf) and BT. Where a BT operator acts as a go-between the Minicom user and the person at the other end using a standard phone. To find out more visit the website at:


<http://www.typetalk.org/>

Case Study: Clare

Clare, who is Deaf, explains how the Internet has benefited her life.

“For me the Internet has opened up a new way to communicate. It has allowed me the freedom to book holidays online, talk to friends via Internet chat rooms and access a wealth of information I never dreamed of a few years ago. It helps me feel part of the wider community.

But I do find it very frustrating that as a textphone (Minicom) user I haven't found many sites that allows me to give my textphone number rather than a phone or fax number as my preferred method of contact.”

 See **Video 6 Sign language and avatars** on the Video demos section of the Resources CD for an example of a website that uses sign language as the main way to communicate.

Mobility impairments

People with mobility impairments face challenges when navigating and interacting with web pages.

They may experience difficulty moving the cursor with the required precision or may lack the manual dexterity or hand-eye co-ordination required to use a standard keyboard. They will often use mouse and keyboard-alternative assistive devices such as trackballs, touchpads, or specialist joysticks.



For those with restricted mobility, voice recognition software such as Nuance's (formerly ScanSoft's) Dragon NaturallySpeaking is often a good solution.

Part one: An overview of web accessibility

Using a headset they give instructions to their computer enabling them to use email, dictate letters and surf the web.



To allow people with mobility impairments to effectively use websites, designers should make sure text links or link images are not too small and such links are not too close together. A typical example of a problem is on a search result page where the links to other pages (e.g. 1 2 3) are too close together. For a good practice example look at Google where the numbers are well spaced out at the bottom of the page.

It is also worth noting that the Windows version of the Opera 8 browser supports voice recognition commands.

 See **Video 8 Voice recognition** on the Video demos section of the Resources CD for an example of voice recognition in action.

Cognitive impairments, dyslexia and learning disabilities

Cognitive impairments include dyslexia, memory impairments and attention deficit disorders.

One of the main issues for this group of users is processing information they see on the screen - any distraction such as excessive movement or poor layout of information impacts on their ability to understand what is on the web page.


To improve accessibility for this group:

- Website content should be organised logically and clearly.
- There should be sufficient spacing between lines, paragraphs or sections and use bulleted lists.
- Graphical icons should be used as navigation aids.

There is also software available for people with reading difficulties such as TextHelp's Read and Write which works with most of the computer programs on a typical PC.

Software for websites includes Browsealoud and ReadSpeaker, they both read out in synthesized speech the contents of web pages.

People with learning disabilities and mental health issues will also benefit from these recommendations.

 See **Video 9 Dyslexia simulation** on the Video demos section of the Resources CD for a practical demonstration of what it is like to have Dyslexia.

Future trends in web accessibility

Making your site accessible to disabled web users has increasingly become a mainstream issue over the last few years, specifically in the UK.

One driver has been the recommendations coming out of the Disability Rights Commission's formal investigation into website accessibility, which puts the onus on web owners and developers to start to make changes in the accessibility of their websites.

Additionally a new guide called PAS 78: 'Guide to good practice in commissioning accessible websites' was launched March 2006 which builds on these recommendations - more about this in the next section on commissioning sites.

Improvements to web design packages

Historically one of the main problems with trying to make sites accessible was the lack of support by the key web design packages such as Dreamweaver, GoLive and FrontPage – leaving accessibility to the domain of hand coders.

This has changed with the new generation of software. For example, Dreamweaver MX has a number of useful accessibility tools built in - however you have to know they are there to turn them on.

Better examples of accessible websites

More and more designers are creating websites to W3C web standards, which means they are more accessible and more compatible with hand-held devices such as mobiles, Palms and Pocket PCs.

More accessible Flash

Adobe (formerly Macromedia) has steadily improved how well Flash works with adaptive technology with each new release. It is not perfect but it has come a long way from the early days of 'total inaccessibility'. The next challenge with Flash is for developers to start using the accessibility features built into Flash to give disabled users a positive experience of the rich media.

Part two: For commissioners

Introduction

Many organisations including those in the voluntary and community sector think making their website accessible will be expensive and therefore will drain resources from other areas. This will not be the case if you take a strategic approach as outlined here, which will enable you to minimise costs and maximise the benefits of making your site accessible for your organisation and the community you serve.

To successfully implement an accessible website you need to think through a number of issues.

Firstly you need to gain an understanding of web accessibility so you can make informed decisions about what you need to do and ask the right questions of the people who work on your website, whether they work in your organisation or are external agencies. This section of the guide is about giving you this knowledge, it will cover:

1. What the law says about web accessibility.
2. Planning for an accessible website.
3. Creating an accessibility policy.
4. Working with designers and developers.
5. Post Launch - how to ensure your site stays accessible.

What the law says about web accessibility

Around the world there are a number of laws that cover web accessibility, here we overview the key laws focusing on the legislation in the UK.

United Kingdom

In the UK the Disability Discrimination Act 1995 (DDA) is the legislation that protects the rights of 8.9 million disabled people in employment and in the provision of goods and services. The requirement to have an accessible website under this legislation has been in force since 1999, but not legally tested – as yet.

In the UK the RNIB will support any individual or group who have a complaint about the accessibility of a website. A number of cases have already been settled out of court and more are likely in the future.

Disability Rights Commission (DRC) Report

Because of a growing awareness that many disabled people were having problems accessing the web the Disability Rights Commission (DRC), the body created to protect the rights of disabled people, commissioned a major research project to investigate the accessibility of 1000 UK websites across all sectors. The finding of the research found that Accessibility was universally poor – 81 % of websites tested failed to meet the basic level of accessibility – W3C priority level 1.

The published report made recommendations to key stake holders including Government, website developers and website owners to take the initiative and improve web accessibility.

Part two: For commissioners

They have clearly stated that if this does not happen within a reasonable time frame then they will turn to legal action. The research also highlighted the eight most common accessibility issues encountered by disabled people on the web which are also covered in this guide.

The DRC report can be downloaded from:

<http://www.drc-gb.org/library/webaccessibility.asp>

USA

In the United States there is legislation in place called Section 508 that already requires websites commissioned by the Federal government to be accessible.

A number of cases have been brought which have tried to clarify if the American equivalent of the DDA called the Americans with Disabilities Act (ADA) covers private websites. The initial test cases such as Robert Gumson v. Southwest Airlines were ruled to be outside the remit of the legislation.

However this position has changed with a new ruling in August 2004 by the New York Attorney General Eliot Spitzer, who has gained an agreement with the travel websites Ramada.com and Priceline.com to make their websites accessible to blind and visually impaired Internet users under the ADA.

“Accessible websites are the way of the future and the right thing to do. We applaud these companies for taking responsible and proper steps to make their websites accessible to the blind and visually impaired,” Spitzer said.

“We urge all companies who have not done so to follow their lead.”

Europe

The European Union has in place a directive on making public websites accessible – ‘European Union: eEurope 2002: accessibility of public websites and their content - com (2001) 529 final’.

It is still in the process of being implemented across member states, but there is a strong likelihood that further legislation or directives will be coming from Brussels which will require public websites to adopt accessibility standards.

Australia

In Australia the first major case of web accessibility was won against the Sydney Olympic’s website in 2000 under the Australian Disability Discrimination Act (1992) because they failed to meet minimum accessibility standards (W3C). The complainant, Bruce Maguire, a Blind Australian, was awarded \$20,000, but far more significant was the publicity it gave to web accessibility standards and the way it opened the way for future legislative action around the world.

Planning for an accessible website

To successfully achieve an accessible website you need to take a strategic approach, specifically you must:

1. **Make sure the managers and trustees of your organisation understand there is a legal requirement to make your site accessible under the Disability Discrimination Act** - refer them to the section on the law in this guide.
2. **Find out how accessible your site is now** by having it audited. You don't need to go to an external agency to do this which would cost money, you can instead use the information in the testing section of this guide which steps you through the process of auditing your site using a range of free tools and techniques. This step is critical, without it you won't be able draw up an objective plan of action.
3. **Decide on what standard of accessibility you want to achieve.** It is a very important to be clear about what standards you want your website to meet. As already mentioned W3C Web Content Accessibility Guidelines level 2 is a standard that many organisations are aiming for over the short to medium term.
4. **Create an internal team or allocate an individual to identify accessibility objectives** - in a small voluntary or community organisation there could just be one or two people responsible for the day to day running of your website. However it is important to engage with people across all the roles that contribute to your website, for example design, content creation or maintenance, to discuss the impact of making your site accessible and what the key objectives are in doing this. This team or individual also needs to draw up an accessibility policy as discussed later in this section.
5. **Make sure that the relevant people in your organisation understand about accessibility** - encourage them to read the relevant sections of this guide based on their job role.
6. **Use the information and advice in this guide** - this guide and the accompanying CD contain everything you need to start improving the accessibility of your site.

PAS 78: Guide to good practice in commissioning accessible websites

Disability Rights Commission (DRC) research on web accessibility found that awareness of the issue is high, but good practice is terribly low.

To tackle this the DRC tasked the British Standards Institution (BSI) to produce a PAS (Publicly Available Specification) on commissioning accessible websites.

The PAS is not 'rules', is not 'new web design guidelines' and it is not 'the law'. It is a useful framework that anyone involved in commissioning a website can use to make sure accessibility is integrated into the website project.

The PAS was launched March 2006 and it:

- Gives advice on creating an accessibility policy.
- Explains how disabled people use websites.
- Overviews web technologies.
- Gives guidance on choosing a web developer (checklist).
- Overviews accessibility testing options.
- Details how to do disabled user testing.

The guide is free and is available for download from www.drc-gb.org

Web Accessibility Policy

A key part of making your site accessible is to set in place an accessibility policy to give all those involved in the creation, editing, testing and updating of your site a clear idea of what they need to do.

The accessibility policy should explain things like what level of accessibility you are working towards and how you have thought about the access needs of disabled people.

The policy should specifically contain the following information:

1. What level of accessibility the site meets - both now and longer term. As mentioned earlier many organisations are aiming for Level 2 of the W3C Web Content Accessibility Guidelines (WCAG).
2. How you have thought about the needs of disabled users accessing the information on your site. For example are you confident they can do the core tasks required to use your site such as searching for information, logging in and contributing to a discussion forum?
3. If you are creating a new site an explanation of how you have involved disabled users through the development of the website.
4. A schedule for regularly checking the accessibility of your site using the tools and techniques outlined later in part 5 of this guide.
5. If any part of your website has specific accessibility issues that will impact on the ability of disabled people to use your site, you must document the problems and explain how you are working towards fixing it within a time frame if possible.
6. For those services that are inaccessible you need to explain how disabled people can access this information or these services via alternative means. This could be via email, phone, fax or Minicom (textphone used by Deaf and hard of hearing individuals). Also make sure this information is easy to find on your website.
7. How you encourage feedback from disabled people on the accessibility of your website and what you do with this feedback.
8. How you have engaged with other organisations or individuals who do work on your website to make sure they know about accessibility - don't assume they are aware of the need to make your website accessible.

In addition it is very important that a summary of your accessibility policy is put on your website. This needs to briefly explain your commitment to accessibility, overview the key features you have implemented and identify any parts of your site you know are inaccessible but are working towards fixing.

Make sure the summary of the accessibility policy is easy to find on your website, do not bury it in the site, have a link in the top or bottom navigation links on your homepage.

A word about Content Management Systems (CMS)

Many sites use some form of content management system for generating web pages and for adding and editing content. For accessibility it is very important that you check out the functionality of your content management system - is it flexible enough that it can be modified to produce accessible pages?

You need to identify early if this is going to be a bottleneck for your accessibility planning. Also it is important that the CMS interface is accessible so disabled employees are able to add content to your website.

You might be in the situation where you don't currently have a CMS, but are looking for one. If this is the case ensure you ask about accessibility and use it as a key criteria to base on which one you choose, whether it is open source ('free') or a commercial product. A useful product to benchmark against is Adobe (formerly Macromedia) Contribute - it costs around £99 for a single license and contains a number of useful features that help keep web pages accessible.

You can download a 30 day trial of Contribute from:

<http://www.macromedia.com/uk/downloads/>

Working with designers and developers

If you use external designers and developers to do work on your website you need to ask them about their accessibility knowledge.

If they say they can produce 'DDA compliant' websites you need to question what they mean by this as legally this has no meaning. As mentioned in the overview of the law no court case on web accessibility in the UK has clarified what the specific accessibility requirement for a website is under the Disability Discrimination Act (DDA).

There is currently no nationally recognised system of accreditation for website developers who claim to create accessible websites that uphold W3C guidelines and specifications. Therefore you need to ask them some key questions to check they have enough knowledge and experience to ensure they can make your site accessible.

You should specifically ask for:

- Examples of previous work.
- References from previous clients.
- Evidence they have a practical knowledge of W3C guidelines and specifications.
- An appreciation of the implications of "The Disability Discrimination Code of Practice (Goods, Facilities, Services and Premises)" 2002 edition - see <http://www.drc-gb.org/uploadedfiles/documents/2008223drccopyrightsofAccess.doc>
- Familiarity with assistive technologies such as screen readers and magnifiers used by people with vision impairments.
- A practical knowledge of PAS 78 - not essential but would be an additional indication they take accessibility seriously.

Post launch – how to keep your site accessible

If you have spent time and effort making your site accessible you need to ensure it stays accessible. A common mistake many organisations make is not integrating web accessibility into their web strategy as a key thing to test and plan for. This frequently means that within a few months of the launch or redesign of a site, accessibility problems start to occur.

To help you avoid this, here are some recommendations for keeping your site accessible.

- Make sure you have work systems in place that make everyone involved in the web integrate thinking about accessibility into their day to day jobs.
- Publicise your Accessibility Policy internally, make sure everyone knows it exists and where to find it.
- Monitor and test your site for accessibility - use the testing techniques explained in part five of this guide, to make sure no accessibility errors are starting to creep in.
- New content - when you plan for a new part of the site, or have external organisations produce content, ensure accessibility is on the agenda at the start of the process.
- New Staff - when new staff join your organisation that do any work on your website, make sure as part of their induction they are given this guide to read or have the basics of web accessibility explained to them.
- Ask for feedback on the accessibility of your site from disabled users, this is often helpful in identifying potential problems.

Part three:

For editors: Writing for the web

This section covers the practical issues of writing for the web for editors or anyone who writes content for your website.

It specifically covers:

- Language.
- Pictures and symbols to help understanding.
- Writing alternative text.
- Hypertext links.
- Headings.

Language

Language and how it is used on web pages is critical for accessibility. You can have a technically perfect accessible website, but if the language it uses is unclear, too technical, or not well laid out there will be major barriers for many people understanding and interacting with the site.

A core requirement under the W3C guidelines is:

**“Use the clearest and simplest language appropriate for a sites content”
[WCAG 14 .1].**

This means you should:

- Write in short sentences, in the active voice.
- State the topic of the sentence or paragraph at the beginning of the sentence or paragraph.
- Make only one key point per paragraph.
- Minimise jargon - keep the text easy to understand.
- Use bulleted lists to break up the body of the text.
- Use white space to lead the user through the page.
- Strive for clear and accurate headings and link descriptions. Review every heading, outline, and menu to see if the crucial words mean exactly what is intended - are there simpler words that would convey the same meaning?

How people read web pages

People read web pages differently to the printed page. Writing for the web is therefore very different from writing for traditional printed media. People tend to scan web pages, rather than read them word by word, and they read web pages on average 25-50 % slower than printed text.

Web pages should therefore be designed to assist this scanning process.

Stanford-Poynter Eyetrack Study

The Stanford Poynter project has been researching how people read news on the web.

The research shows that web users:


- Look more at text than at graphics.
- Are first drawn to headlines, summaries and captions.
- Frequently do not look at images until the second or third visit to the page.
- Prefer simple direct headlines that directly reflect the text.

This contrasts with a previous study carried out by Poynter into how people read printed newspapers which found that people tended to be drawn to the photos before looking at the text.

See: www.poynter.org/eyetrack2000

Pictures and symbols to help understanding

As well as keeping your language as clear and simple as possible, it is also important that you use pictures and symbols to aid understanding of page content and give them descriptive alternative text also known as 'alt text'.

This is particularly important for people with learning disabilities and dyslexia, who will often struggle to read pages of text, but will more easily recognise symbols and icons. An example of this would be the use of an  envelope icon for your 'Contact us' page.

The BBC accessibility help site, My Web My Way makes extensive use of icons for each section to help guide people through the pages.

See <http://www.bbc.co.uk/accessibility>

Some people with learning difficulties find the complexity of many websites makes them hard to understand and makes it difficult for them to find what they want.

While people who create websites should keep the language and navigation as simple as possible, and add easy to recognise icons to help people with learning difficulties, this might not always be enough to make the page understandable.

Fortunately there are technological solutions like Symbol browsers which can help individuals further by converting text to easily recognised symbols. An example Symbol browser is Communicate: Webwide by Widgit (it used to be known as 'Webwise') which helps people with learning difficulties in two ways:

- Non-readers can access information using only symbols.
- For those with some text knowledge, the symbols act as a reminder for the words that are not immediately recognised.

See: <http://www.widgit.com/>

Writing alternative text

For accessibility all non-text elements on a page such as images and Flash movies need alternative text often known as 'alt text', so that screen readers and other adaptive technology can understand what is on the page.

Equally important as having alt text is the way that it is written, as a guide alt text should:

Be concise: unnecessarily long and detailed alt text is time-consuming to read and impairs effective site navigation - avoid clutter.

Be meaningful: vague and non-specific alt descriptions do not provide sufficient information to users who cannot see the non-textual items. Avoid text such as 'presenter's photo' or 'click here', and replace them with more precise descriptions such as 'Alan Titchmarsh' and 'this season's garden tips'.

Replace the purpose of an image: alt text should be used to replace the purpose of an image, what is important or relevant about the image in its context, not to simply provide a label or description.

Not use redundant words: text such as 'image of', 'picture of', 'photo of' are unnecessary because adaptive technology such as screen readers tell the user that it is an image or non text object.

Also if an image is essentially functional, for example a "Search Now" or "Go" button, its alt attribute should convey this – a description of what the button looks like is unnecessary.

Non text elements that do not provide information

Elements that do not provide information about what is on the page should have an "empty" alt text description. For example if an image is essentially 'window dressing', or is used solely for the purpose of visual layout (e.g. transparent 'spacer' images, background images or other furniture) its alt-attribute **must** be empty.

For example: ``

Images as links

If an image is used as a link, its alt-attribute **should** describe the destination of the link. For example, if an image on a home page links to an article, the alt attribute should be the title of that article for example you should use "Review of The Matrix" rather than "Neo fights Agent Smith".

If you feel that a description of the image is also useful, you should add this information to the end of the alt attribute. For example "Review of The Matrix (Image: Neo fights Agent Smith)".

You should be cautious about including words like 'link to' and 'jump to' in alt text for link images. You do not need to start your description with these phrases because users are normally already prompted with this information by their assistive technology, for example a screen reader often changes voice, from male to female to indicate links.

Hypertext links

In order to be understandable and usable regardless of impairment, text links have four key requirements:

- The text must be legible and selectable.
- It must be clear that the text is a link (links must be self-evident).
- The text must be self explanatory, for example do not use 'click here'.
- The same link phrase should not be used for links that go to different pages. For example don't use 'more news' twice with each link going to a different news story - make each link text description unique.

In general, your links design **should** balance aesthetic qualities with the constraints set by usability, accessibility and browser degradation.

The standard blue underlined links have some usability issues. Specifically they cause legibility problems by interfering with descenders on letters such as 'g', 'y' and 'j'.

Also the human eye, especially in the case of older users, is generally not well suited to detecting blue.

As a general trend mouseovers are emerging as a new standard to indicate links. A mouseover is when you hover over a link and an element of the link such as the background colour changes to indicate the hover state. This can be seen on Windows XP menus, many websites and Flash movies. For people who use the keyboard to navigate the web, sites are also starting to use techniques to make the links stand out, for example go to www.msn.com and tab through the links.

Make links easy to access

A text link **should** have sufficient space around it for the user to be able to accurately select the link, particularly where several links appear together.

These requirements are especially important to users with vision problems and those with motor control problems who struggle to click on small areas.

Make links obvious

You should use a CSS mouseover colour change or underline, and either a markup or context cue to indicate that a piece of text is a link, regardless of platform and user preferences:

Markup: use font-weight or font-size styling options (Cascading Style Sheets) to distinguish links from surrounding text.

Context: position links in specific areas of the pages; bullets or appropriate graphics can be also be used to indicate a list of links.

You may also use a different colour for the link from that of the surrounding text as an additional cue. Colour is seen as an additional, rather than a primary cue because some people may have difficulty distinguishing between colours.

Reasoning

Two cues are necessary because a CSS mouseover colour change will not show if users have specified their own CSS style sheet, disabled CSS, or are using a browser with no support for CSS. You **should** use colour to distinguish between visited and unvisited links – bearing in mind possible colour blindness issues.

Headings

Clear headings will trigger a user's interest and draw them to a relevant section of the page. They will also help them quickly overview the information available on the page.

It is therefore important that headings are legible and understandable for all.

Humour or confusion?

Some users, especially those with low level of understanding of double meanings, will feel deceived if a heading seems to promise something it fails to deliver. Caution must be adopted with respect to short, punchy "straplines" that can be interpreted in several ways. Better to be less witty than confuse readers.

The problem with ALL CAPS

Most of us, but specifically many people with dyslexia, recognise words by their shape, not by deciphering the exact letter and syllable combination.

A sentence or group of words written in ALL CAPS is cumbersome to read for most people – it can become totally impossible for web users who cannot read by any other means other than word shape recognition.

Therefore, it is better to emphasis headings via bold fonts, bigger font size or different colour, using CSS style sheets rather than writing them in all caps.

Editors checklist

Below are some key checkpoints to consider when writing content for the web:

- Use the simplest and clearest language appropriate for a site's content.
- Use short line lengths and paragraphs.
- Use pictures and symbols in addition to text.
- Provide a text equivalent for each non-text element / ensure pictures have descriptive text (alt attributes).
- Avoid putting 'too much' information on a page - as a guide don't have more than three pages worth of content as you scroll down.
- Ensure there is suitable space around links and make sure that links are large enough to select easily.
- Link text should contain information about their destination - don't use 'more' or 'click here'.
- Make sure you do not use the same link phrase for links that go to different pages - make each page link description unique.
- Is the meaning of each page and section heading clear?
- Have you used 'Title Case' or 'Sentence case' rather than 'UPPERCASE' for headings?

Part four:

For developers and designers

This section is for developers and designers and explains all the key accessibility issues and the best techniques for dealing with them.

Please note that not every accessibility issue listed in the W3C's Web Content Accessibility Guidelines (WCAG) checkpoint list is explained in this guide. However all the most important level 1 and level 2 issues that will make the most difference to the accessibility of your site are covered? Important checkpoint references are given in the text as [WCAG x.x] - there is a complete levels 1 & 2 checkpoint list in the resources section in part 6.

Web accessibility in context

“...99.9 % of all websites are obsolete” [because they don't use web standards]

Jeffrey Zeldman,

Internationally renowned web expert and author of 'Designing with web standards'.

Web accessibility should not be treated in isolation; it is part of the wider move to designing with web standards. Web standards cover:

Structure: the way your pages are coded using html/xhtml/xml web languages.

Presentation: how they look, typically this is done with style sheets, for example Cascading Style Sheets (CSS) versions 1 and 2.

Behaviour: how scripting is used to maximize compatibility across browsers and be as accessible as possible.

The benefits of web standards include:

- Reducing the cost and complexity of web development.
- Making your sites backwards and forwards compatible.
- Making your site work effectively with adaptive technology.

Web accessibility level I issues

This part of the guide looks at the main level one accessibility issues, which in summary are:

- Colour.
- Images.
- Inaccessible Flash.
- Cascading Style Sheets (CSS).
- Data Tables.
- Frames.
- Multimedia.


The reason for each issue is discussed and the recommended solution is explained.

Colour

Issues with using colour cut across all the accessibility levels. Under the level I guidelines the most important issue is not using colour alone to convey information. For example 'click on the green button to start' or 'click on the red button to stop'. **[WCAG 2.1]**

The reason for this is that people with colour blindness and other vision problems cannot easily distinguish between certain colour combinations. For example one of the most common is being unable to tell the difference between red and green.

So in the 'button' example above many people will just see two different shades of brown. The best solution is to convey information by using non colour specific language, such as click on the 'Go' button to start (which can be green) and click on the 'Stop' button to stop (which can be red).

 See **Video 1: Colour blindness** in the Video demos section of the Resources CD for a practical demonstration of this issue.

Images

The use of alternative text also known as 'alt tags' for all graphics on a page is fundamental to accessibility. It is responsible for around 30-40 % of all problems affecting a range of disabled people accessing the web. **[WCAG 1.1]**

All graphics on a page need to be labelled because a blind person using a screen reader has only the information in the alt tag to gauge the importance of a particular image. In addition it impacts on voice recognition users who rely on navigating a website by voice commands – if alt text on graphical links or form buttons are missing, or not intuitive then it will impact on their ability to interact with the site.

Also some people with dyslexia use software such as Texthelps Read and Write to speak the content of a page outloud and this includes pictures and graphical links. So if alt text descriptions are missing they will have problems understanding and reading the page.

The range of images that need to have alternative text, include:

- Pictures, logos and graphics that complement text.
- Graphical links.
- Graphical form buttons.
- Spacer/decorative graphics.
- Graphs and charts.
- Images maps.

Note: See the section 'Writing alternative text' in Part three of this guide for best practice guidance on writing alt text.

Pictures, logos and graphics that complement text

For logos, pictures or graphics that complement text you add an alt tag with a terse description of the image, for example the Citizens Advice logo is coded as follows:

```

```

Graphical links

If the image is a link, you must describe the destination or purpose of the link not a description of the image itself. Please note text such as "click here for..." is unnecessary in the alt text of a graphical link.

Example:



```
<a href="/products/acrobat/readstep2.html">
</a>
```

Graphical Form buttons

Typically these are used for submit and search buttons on websites, as with graphical links the alt text should be descriptive of the function of the button for example:

```
<input type="image" src="/images/srchb2.gif" name="go" value="go" alt="Go" width="64" height="25" border="0" />
```

Spacer/decorative graphics

For spacer gifs or decorative images you use a null or empty alt tag (alt="").


This is so screen readers can ignore the image which provides no useful information for a blind web user, for example:

```

```

Important!

Do not put empty alt tags (alt="") on either images that convey useful information or image links because it causes serious problems for screen reader users. Not adding an alt text means a screen reader will use the url (web address) of the link instead, which will often not make sense.

 See **Video 4: Screen reader - images** on the Video demos section of the Resources CD for an example of a web page with link images missing alt text.

Graphs and charts

If the image is a chart or graph then you must provide a longer description than the "alt" attribute can reasonably display. The "longdesc" attribute was created for this reason.

Example code:

```
<IMG SRC="images/upchart.jpg" width="372" height="409" align="left" border="0" longdesc="benefitschart.htm" alt="chart of strategic benefits">
```

Where benefitschart.htm would contain a detailed text description of the chart.

Image maps

If you use image maps on your site, make sure that each active region has an alt text description. For example:

```
<map name="top_nav_map">
<area shape=rect href="about.htm" coords="0,0,99,22" alt="About us">
<area shape=rect coords="99,0,162,22" href="accessibility.htm" alt="accessibility help">
<area shape=rect coords="162,0,263,22" href="sitemap.htm" alt="site map">
<area shape=rect coords="263,0,300,22" href="contact.htm" alt="Contact us"></map>
```


Flash

Flash has a reputation for being one of the most inaccessible web technologies, however accessibility is slowly improving with each new version of Flash. Unfortunately the new accessibility features only work with newer versions of four screen readers: Window-Eyes, Jaws, Hal and IBM's HomePage Reader 3.04.

Because of the cost of upgrading screen readers it will take time before the majority of visually impaired/blind users will have Flash compatible screen readers.

Currently for accessibility, Flash should be used to enhance an html page not replace it.

Always provide an accessible alternative to the Flash content, for example if you use a flash menu provide an alternative accessible html version of the menu at the bottom of the page. **[WCAG 1.1]**

 See **Video 5: Screen reader - Flash** on the Video demos section of the Resources CD for a practical demonstration of how inaccessible Flash can be.

Flash accessibility for current screen readers

To make your Flash accessible to modern screen readers you need to follow best practice guidance on accessible Flash from Adobe (formerly Macromedia). In summary:

- All graphical elements should have text equivalents.
- Control reading order of the Flash content (make sure the reading order makes sense to someone who cannot see the screen).
- Caption audio content.
- Make looping elements inaccessible (this is so screen readers don't become stuck in a continuous loop).
- Allow users to control motion (for example turn it on or off).
- Ensure keyboard access to all controls.
- Expose structure of complex Flash movies (helps screen reader users orientate themselves in the movie).
- Expose state of controls (for example is the control on, off or in another state?)
- Use colour wisely (make sure there is good colour contrast).
- Check for accessibility (use a screen reader such as Jaws to test your Flash).

For more information see the 'Best practices for accessible Flash design' guide in the links section on the Resources CD.

Cascading Style Sheets

Cascading Style Sheets (CSS) allow you to separate structure (html) from presentation (design) on your web pages. It enables you to:

- Control features such as text alignments, fonts, colours and background images.
- Provide templates for laying out your page, helping you to maintain consistency.
- Use much cleaner, clearer page code - therefore more accessible page code.
- Save time in page redesign.

However for level 1 accessibility, if you use CSS you must ensure that your pages work without it. For example if you set your font colours in a style sheet you must also set your background colours in the same style sheet rather than in the page code. This is because if the link to the style sheet is broken for any reason then your pages might become inaccessible due to a mismatch of text and background colours. **[WCAG 6.1]**

You can provide CSS styling for older browsers (CSS1) by hiding the more advanced CSS (CSS2) using the @import command rather than using browser detection scripts.

Netscape 4.x and Internet Explorer 4.x don't recognise the import command and therefore ignore the advanced style sheet meant for more up-to-date browsers. In the example code below the version 4 browsers will only detect the basic.css style sheet, while more recent browsers will also import the advanced.css style sheet.

```
<LINK REL="stylesheet" HREF="basic.css" TYPE="text/css"><STYLE TYPE="text/css">
<!-- @import url(advanced.css);--></STYLE>
```

You can also specify a print style sheet that optimises your pages for print by using the media="print" option as shown below. It only works in browsers such as Internet Explorer 5.x. and 6 and Firefox.

```
<STYLE TYPE="text/css" MEDIA="print">
<!-- @import url(print.css);--></STYLE>
```

Scripts - JavaScript/Applets

A level 1 requirement is that your web pages work if JavaScript and other scripts are turned off or not supported. **[WCAG 6.3]**

JavaScript is one of the most common sources of accessibility problems. This is because many older browsers have no or limited script support. Around 5% of people will not be able to use a site that relies on JavaScript, either because they use browsers that do not support JavaScript or have it disabled.

Source <http://www.thecounter.com/stats/2004/May/javas.php>

For example you cannot log into your Hotmail account if you turn off JavaScript in your browser.

So as a main recommendation whenever possible, use server-side scripting such as php or asp rather than client-side scripting such as JavaScript.

If you must use JavaScript, for minimum accessibility:

- Don't use JavaScript that requires a mouse for it to work (however graphical rollovers are not a problem).
- If you use JavaScript functions in links, also provide a real (html) link destination.
- Avoid the use of pop-ups to give important information.
- JavaScript should never be relied upon for essential functionality such as navigation.
- Test your pages with JavaScript turned off (for example use the Firefox browser).
- Provide meaningful <noscript> content, offering access to a non JavaScript version with the same functionality – but don't assume a screen reader will pick this up, many support JavaScript so it must be as accessible as possible.

The most common JavaScript accessibility issues include:

Navigation: inability or difficulty navigating using a keyboard or assistive technology.

Hidden content: presentation of content or functionality that is not accessible to assistive technologies.

User control: lack of user control over automated content changes.

Confusion/Disorientation: altering or disabling the normal functionality of the browser or triggering events that the user may not be aware of.

For more information on JavaScript see AbilityNet's 'Advanced accessible JavaScript' PowerPoint presentation which is available at: <http://www.abilitynet.org.uk/content/oneoffs/pre-tech.htm>

Tables for data

The main reason to code tables of information or 'data tables' correctly is to enable blind web users who use screen reader technology to make sense of the information. It enables them to go through a table item by item and at a key stroke check the name of the row or column they are in.

[WCAG 5.1]

Particularly important is that column headings are marked up with the table header tag <th>, rather than the standard table data <td> tag.

In addition to using the correct tags for column headings there are a number of enhancements you can make to the code to make them more screen reader friendly. The most important of which is called the 'scope' attribute, which explicitly associates data cells (coded in <td> tags) with their related column and row headings to give screen readers more information about the table structure.

The Scope attribute can have any of the following values: "row", "col", "rowgroup", or "colgroup."

Below is example code snippet for a football league results table:

```
<tr><th scope="col">Team</th>
<th scope="col">Played</th>
<th scope="col">Goal difference</th>
<th scope="col">Points</th>
</tr><tr>
<td >Reading</td>
<td >41</td>
<td >61</td>
<td >95</td>
</tr> -- // --
```

There is an alternative approach which is more involved using the 'headers' attribute which is better suited to more complex tables.

See <http://www.w3.org/TR/WCAG10-HTML-TECHS/#tables> for more details.

Frames

Sites with Frames and iFrames cause barriers for adaptive technology users - specifically navigating around a site is very difficult if frames are not titled clearly and accurately to their purpose and content. This is because screen readers and text browsers view a web page built with frames not as a complete page but as a list of frame links, so they have to navigate into each frame to view its contents.

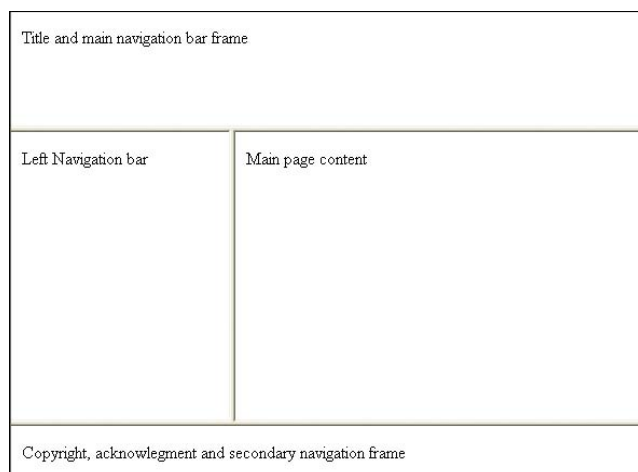
[WCAG 12.1]

Often they will have two or more choices, which is why meaningful frames titles should be used so they can easily identify the frame they want to open.

You add a title to each frame as shown in the code snippet below.

There are four frames in this example. The frame along the top of the page is for the page title and primary navigation bar. The narrow frame (centre, left) is for the left navigation bar. The largest frame (centre, right) is for the main page content. The frame across the bottom of the page is for the copyright, secondary navigation and other constant information.

```
<FRAMESET ROWS="20%,*,30%">
<FRAME SRC="f1.htm"
title="Title and main navigation bar">
<FRAMESET COLS="20%,*,">
<FRAME SRC="f2.htm" title="Left navigation"> <FRAME SRC="f3.htm" title="Main page content">
</FRAMESET>
<FRAME SRC="f4.htm" title="Footer - Copyright, acknowledgment and secondary navigation frame">
</FRAMESET>
```



Multimedia

Multimedia accessibility is important for both hearing impaired and blind people. The most important issue is to provide captioning options for video or at the very least provide transcripts that can be read as an alternative. **[WCAG 1.3, 1.4]**

Audio clips or speech should have alternative descriptions and/or a text transcription to make them accessible to people with hearing impairments.

Video clips should have what are known as ‘closed captions’; Windows media player, QuickTime and Real player all have these options. The methods and techniques for each do vary, fortunately the Webaim site has excellent tutorials on captioning multimedia which can be found at:

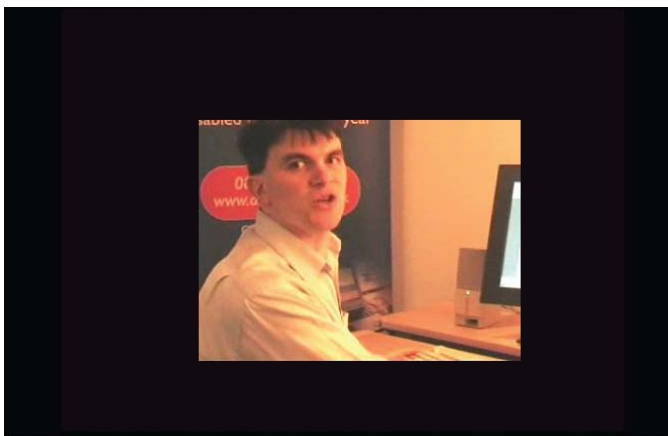
<http://www.webaim.org/techniques/>

Also the National Center for Accessible Media (NCAM) provides examples of accessible multimedia and a free tool called MAGpie which adds captions to video. More details are available on their website at:

<http://ncam.wgbh.org/richmedia/>

Best Practice

- Provide a link for users to download the latest version of the required media player.
- Provide an accessible text transcript of the presentation.
- Ensure captions are synchronised with the timing of multi-media content.
- Provide controls that allow users to control the multimedia presentation such as ‘stop’, ‘play’, and ‘pause’ if you embed the video in the page.



An example of captions in Windows media player.

Key Level 2 accessibility issues

This part of the guide looks at the main level 2 accessibility issues.

Not all organisations will be in a position to be able to tackle the range of issues discussed here. Until you have a good understanding of level 1 issues, and have implemented the necessary changes to your website don't try and do too many level 2 accessibility fixes.

In summary the key level 2 accessibility issues are:

- Hypertext links.
- Relative design.
- Device independence.
- Mark-up (X/html).
- Colour contrast.
- Movement.
- Forms.

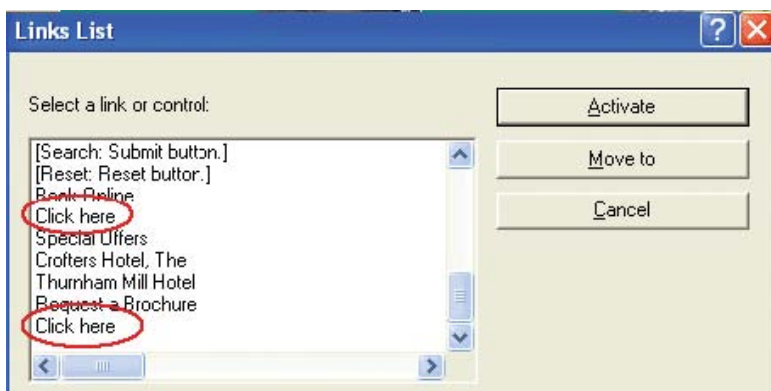
Hypertext links

Clear and concise hypertext links are key to accessible website navigation.

The main accessibility issues for hypertext are:

- Link text that does not make sense out of context such as 'click here'.
- Duplication of the same link text such as 'more information' which point to different web pages.
- Non-standardisation of links across a site such as using different colours for links on different pages.

Screen reader users often create a list of links from a page to help them navigate more easily. When you have a link description such as 'click here' or 'more' taken out of context it does not make sense, and this is compounded when you have multiple links with the same description. It also makes life easier for voice recognition users who say 'what they see' to navigate web pages.



Link best practice

- Have a terse but meaningful link description – unique to each link on a page, do not use ‘click here’ or ‘more’ for example. **[WCAG 13.1]**
- When you link out of your site to another site or a link opens a new browser window you need to give some warning. **[WCAG 10.1]**
- Use the link title attribute to expand the description of a link giving the user more information about whether they want to follow the hyperlink.
- Avoid pop up windows and ‘spawned’ windows. **[WCAG 10.1]**

It is also a good idea to differentiate between structural navigation links such as top and side navigation bars from the links you have in the main content of a page.

Relative design

A level 2 accessibility requirement is that page layout dimensions and font sizes are relative, in percentages or ems (a unit of width relative to a font size).

This means web pages are automatically scaled when you change the browser window dimensions. It allows the text to change size and the page layout to flow without creating extra whitespace or running off the edge of the screen.

This is particularly helpful for vision impaired web users who need to be able to read the page text in a larger font.

If you use tables for layout then use percentages values for widths and heights for table cells, columns and rows to create relative layouts. **[WCAG 3.4]**

 See **Video 2: Changing text size and background colour** in the Video demos section of the Resources CD for a practical demonstration of issues with fixed font sizes.

CSS positioning for layout

More and more mainstream sites are now starting to use CSS positioning for layout.

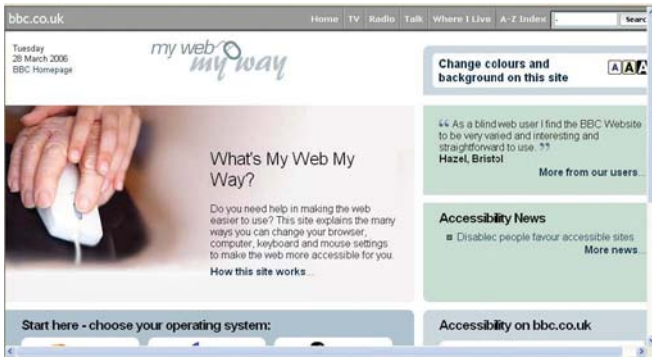
While W3C web accessibility guidelines recommends CSS for layout it does take a lot of trial and error to find a design robust enough to work well in a range of browsers. So as a first step many organisations are using a hybrid of a simple table layout with CSS for their page designs.

However using CSS for layout does have its problems, for example:

- Long line width on wide screen monitors - which is not good for readability.
- Lack of control of design elements. For example often changing the text size will break the page layout.

One option is to use the “em” a relative unit of measurement to create a compliant “fixed” width CSS layout where the page will resize proportionally and eliminate the above problems.

For example go to www.bbc.co.uk/accessibility and increase the text size.



However even this solution is not perfect, it causes horizontal scrolling on 800 x 600 monitors.

If you want to know more about designing your website with CSS positioning see the list of recommended books on the Resource CD.

Device independence

Your website should be “device-independent accessible” [**WCAG 9.3**] so that users can interact with your website if they use a device other than a mouse. For example via voice recognition, keyboard or a head wand. If your website can be navigated via the keyboard then other input devices should work.

Designing for device-independence will require you to test your website with the keyboard and other mouse alternatives through the design process. Additionally many browsers such as Internet Explorer, Firefox and Netscape contain accessibility features that enable users to use keyboard shortcut keys called accesskeys to more efficiently navigate websites. For an example go to www.bbc.co.uk and while holding down the ‘Alt’ key press ‘0’ to bring up a list of accesskeys used on the BBC website.

If you use Flash movies embedded in your page make sure that people can use the keyboard to both navigate through any Flash links and select them.

JavaScript and device independence

When JavaScript is used to trigger specific events it should be device independent. In other words it should not require the use of a mouse. [**WCAG 6.4**]

The onmouseover event handler causes the most accessibility problems. As its name implies, onmouseover requires the use of a mouse for an action to happen such as a dynamic text pop-up - screen readers or keyboard users have no way of accessing this information unless an accessible alternative is provided. Onmouseover, and its companion, onmouseout, can be used however as long as any important content or functionality is also available without using the mouse.

OnClick is technically a mouse specific event handler but in practical terms it is device independent. If it is used with hypertext links or form controls, then nearly every browser and assistive technology will work with pressing the ‘Enter’ key to trigger the action.

For more information on device independence see:

<http://www.webaim.org/techniques/javascript/eventhandlers>

Mark-up (X/html)

A key requirement of level 2 accessibility is that the languages you use to create your web pages, such as html or xhtml should be coded properly. So for example headings, lists, and quotes should be marked up correctly with the relevant code tags.

Another important concept is that your pages validate against the standard you specify at the start of your web page code. This is known as a doctype definition or DTD.

Headings

A screen reader user can extract a list of headings from a web page to gain an overview of its structure, similar to the outline view in word if the headings are coded accessibly. This gives them a quick way to skim the page content rather than plough through the whole web page line by line, which is more time consuming.

To code headings accessibly you use the header tags, such as H1, H2, and H3 in the html code rather than graphics or CSS formatted table tags for headings. **[WCAG 3.5]**

For example a CSS style heading such as:

```
<TD vAlign=top width="130">  
<SPAN class="titleblue">  
Welcome to our VCS website</span>
```

Should be coded as:

```
<H1 class="titleblue"> Welcome to our VCS  
website</H1 >
```

Each level of heading has a different weight with h1 as the most important and h6 as the least important. As far as possible use the h tags to add structure to your page. Also it is good practice to only use one level 1 heading on a page and try to 'nest' the headings so h3 follows an h2 heading, rather than an h4 heading following an h2 heading for example.

Lists

All the types of lists you use on your site, including bulleted and numbered lists, should be coded using the correct html tags rather than inserting graphics for bullet points or just typing in the numbers for numbered lists. **[WCAG 3.6]**

To create standard bulleted or unordered lists use the following code:

```
<UL type="disc" >  
<LI>Point 1 </LI>  
<LI>Point 2</LI>  
<LI>Point 3</LI>  
<UL>
```

The type="disc" is optional as a black disc (circle) is the default bullet style. However you can substitute "square" for "disc" for square bullet points and alternatively "circle" for open circle bullet points.

For default numbered ordered lists use:

```
<OL>
<LI>Item 1</LI>
<LI>Item 2</LI>
<LI>Item 3</LI>
</OL>
```

Note: you can turn the above ordered list into an alphabetical list (e.g. a., b., c.) by changing the tag `` to `<OL type= "a">` and to a Roman numeral list by changing `` to `<OL type="i">`.

Graphical bulleted lists

If you want to add graphical bullets the best way to do it is using Cascading Style Sheets (CSS).

For example to produce the following effect:

- Item 1
- Item 2
- Item 3

Create the following html code:

```
<UL class="bullet">
<LI> Item 1 </LI>
<LI> Item 2 </LI>
<LI> Item 3 </LI>
</UL>
```

And add the following CSS code in your style sheet:

```
UL.bullet {list-style-image: url("images/bluebullet.gif"); list-style-type: disc;}
```

Note it is good practice to set the default bullet type (disc) just in case the graphic bullets don't load.

You can also turn off the bullet effect with CSS. For example you can turn off bullets for list items nested within another list using the following style:

```
UL UL {list-style-image: none;}
```

Another increasing common use of this technique is to use lists for site navigation to maximise accessibility.

Avoid using unsupported html tags

You should avoid old features of html such as the font tag which are no longer supported by current web standards. They can cause unpredictable behaviour with a range of adaptive technology used by disabled people. **[WCAG 11.2]**

The font tag should be replaced by coding in your style sheet (CSS) which is a more flexible way to set font attributes such as type, colour and weight.

Quotes

There are two types of tag for presenting quotes correctly, <BLOCKQUOTE> and <Q>. The <Q> tag is used to identify speech dialogue. The examples below show how they should be used:

```
<BLOCKQUOTE>&quot;The power of the web is in its universality. Access by everyone regardless of disability is an essential aspect.&quot;<BR>
```

```
<EM>-- Tim Berners-Lee, W3C Director and inventor of the World Wide Web</EM>  
</BLOCKQUOTE>
```

```
<P>
```

```
----<Q> &quot; My Name is Chris &quot; </Q> he replied----
```

```
</P>
```

Note ‘"’ is the correct way to code quotation marks in html. **[WCAG 3.7]**

Page validation

For level 2 accessibility you need to ensure your pages validate to formal grammars. **[WCAG 3.2]** In practical terms this means including a document type declaration (DTD) at the beginning of web pages, and checking your page code with html/xhtml validators to make sure you adhere to the rules of the published DTD that you are using. For example the transitional html 4.0 DTD is coded as:

```
<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.0 Transitional//EN”>
```

Having a DTD helps ensure consistency in web documents and enables devices such as screen readers to know what standards have been used on the page so they can interpret them correctly.

You can use the following or similar tools to identify where the code errors are in your pages and fix them.

X/html validator at: <http://validator.w3.org/>

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

It is worth knowing that the new version of the W3C Web Accessibility Content Guidelines 2.0 currently out in draft places much more emphasis on validation of code, and it is possible it will change to a level 1 accessibility checkpoint.

Additional recommendations for accessible coding

- Use the <P> tag for paragraphs.
- Use the end tag, for example close a paragraph with a </P> and a list item with a .
- Use the tag rather than the (bold) tag and use the Emphasis tag rather than the <I> (italic) tag.
- Avoid using tags as much as possible - it will make your page code easier to read.

Colour Contrast

Ensure there is sufficient contrast between text and background colours you use on your web pages. For example do not put white text on a pale or pastel background, or black text on a dark colour background. This is level 2 issue for graphic text and level 3 for text. **[WCAG 2.2]**

You can use software such as the AIS Colour Contrast Analyser tool to check for colour contrast. You can find out more about it in the Accessibility tools section on the Resources CD.

For people with colour blindness ensure you check your website colours with either the Vischeck on-line page checker or Photoshop plug-in available at: <http://www.vischeck.com/>

Movement

It is a level 2 accessibility requirement to allow the user to stop movement on a page. **[WCAG 7.3]** Movement is a distraction to people who have cognitive or memory impairments and it is particularly an issue for individuals who suffer from photosensitive epilepsy who can have seizures triggered by movement on the screen.

It is specifically important for level 1 accessibility to ensure that you do not cause the screen to flicker **[WCAG 7.1]** and for level 2 you do not use blinking images or text. **[WCAG 7.2]**

One solution to minimise movement on your pages is to re-program animated gifs and Flash movies to run for a limited number of frames or pause movement by putting in a rest phase within the animation cycle. Alternatively provide a scripted button which can turn off the page movement with one button click/press. It is worth remembering that there is an option in Internet Explorer to turn animated gifs off in the preferences menu and by pressing the 'Escape Key'.

One of the findings of the Disability Rights Commission's formal investigation into website accessibility, mentioned earlier in this guide was that many dyslexic people found animations very distracting and would prefer to be able to switch them off.

Forms

Coding forms correctly is important because blind web users accessing the web with screen readers need to know what input boxes are for. If they are not labelled properly it is confusing and often hard to identify what a particular field for.

The two specific issues are:

- Using the label element to explicitly associate a form field with its description. **[WCAG 12.4]**
- Ensuring that labels of all form controls are properly placed – for example radio buttons and checkboxes have their label text on the right hand side which is the opposite of all other form fields which have their label text on the left. **[WCAG 10.2]**

Label for radio button goes here.

Label for checkbox goes here.

Part four: For developers and designers

A simple example is given below of how to accessibly code form input elements. Note the value of the “for” attribute must be the same as the value of the “id” attribute.

```
<LABEL FOR="name">Name:</LABEL>
<INPUT type="text" id="name" SIZE="50"> <BR><LEGEND>Sex</LEGEND>
<INPUT TYPE="radio" name="sex" id="male"> <LABEL FOR="male">Male</LABEL>
<INPUT TYPE="radio" name="sex" id="female"> <LABEL FOR="female">Female</LABEL>
<BR><LABEL FOR="work">Type of work:
</LABEL>
<SELECT NAME="work" id="work">
<OPTION>Full time</OPTION>
<OPTION>Part Time</OPTION>
<OPTION>Voluntary</OPTION>
</SELECT>
```



Name:

Sex Male Female

Type of work:

Additional level 2 accessibility issues

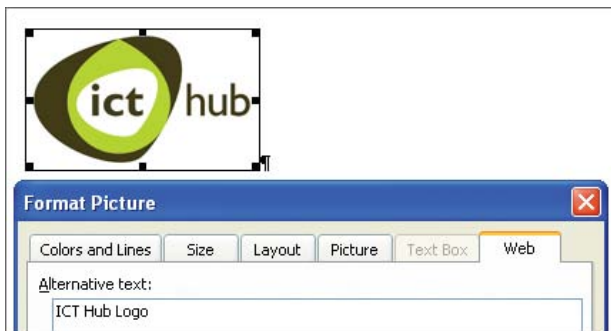
- Include metadata in your web pages. [WCAG 13 .2]
- Avoid auto refreshing pages. [WCAG 7.4]
- Do not use markup to automatically redirect pages – configure the server to do it or provide a manual link. [WCAG 7.5]

Word, PowerPoint and PDFs

On a website you will often find a range of documents you can download in various formats such as Word, PowerPoint and PDF. Unfortunately frequently these documents have not been created with accessibility in mind. This section briefly outlines key things to think about when you create these documents.

Word documents

- Use structural mark-up such as heading styles and bullets in the documents - it will help screen reader users understand the documents and will make it easier to convert them to PDFs.
- Add alt text descriptions to images and ensure that for charts and graphs the captions with the diagrams are descriptive enough to make sense by themselves.



- Ensure that heading are clear and relate to the following section text.
- Use the simplest language possible for the document.

PowerPoint

- Use the auto layout templates supplied with PowerPoint.
- Avoid the temptation to add in lots of text boxes to slides. This can cause problems with the reading order for screen readers as they read them in the order the boxes were created not by their position on the screen.
- Add in alternative text descriptions for key images and add in extra descriptions in the notes pane at the bottom of the page.
- Use a clear font and make sure it is readable from a distance.
- Ensure there is sufficient contrast between text and background colours on the slides.
- Caption any multimedia you use.
- Think about the language you use in your presentation, is it understandable to the widest audience?
- Avoid too much distracting movement on a slide.

PDF

PDF accessibility for screen readers requires that:

- The document must contain real text (not an image of text).
- The document must be in tagged PDF format.*
- The document must be marked up for accessibility.*

(*Only possible in Acrobat 5.0 and above.).

There are three ways to generate tagged PDF:

- Use a Microsoft Office product to generate the document such as Word then convert it to PDF via PDF writer software.
- Run the “Make Accessible” plug-in on existing PDF documents.
- The hard way: create the tags yourself.

To find out more see the following documents in the links section of the Resources CD:

- [Creating accessible PDF files a guide for document authors version 6.](#)
- [Reading PDFs for people with Disabilities version 6.](#)
- [Reading PDFs for people with Disabilities version 7.](#)

Part five: Testing the accessibility of your website

Testing overview

It is important to test your websites and web applications under a number of conditions so that you replicate the different ways people will try to access your website .

You can check how accessible your site is in a number of ways:

- Web Audit against W3C guidelines.
- Standard browsers/different operating systems.
- Test with the relevant inbuilt accessibility features of browsers and operating systems.
- Testing tools: there are a number of automated tools that you can use to test your website such as WebXact (Replaces 'Bobby').
- Adaptive technology – use screen readers, voice recognition, magnifiers for example.
- Disabled user testing - it offer a deeper level of insight into your site's usability and accessibility.

As having your site audited by an external organisation could well be unrealistic, we have put together a set of testing tools that are free and that cover all the main testing requirements. The rest of this section overviews the tools and gives you practical guidance on testing the major accessibility issues.

Your free tool kit for testing

- Automated testing tools such as WebXact.
- Demo versions of screen readers.
- Internet Explorer.
- AIS Toolbar.
- Opera browser.
- Text-only browser such as Lynx.
- JuicyStudio.com.
- Vischeck.
- Your keyboard!!!! (Without a mouse).

Automated testing tools

Below are some of the most commonly used online automated testing tools you can use to check your web pages for accessibility:

WebXact

<http://www.webxact.com/>

Accessibility is only part of a wider range of checks this tool can do, it also does quality checking such as testing that all your web links work.

The tool is good but not perfect:

- There is quite a lot to understand - you need a good knowledge of the W3C Web Content Accessibility Guidelines to use it.
- As it is popular it can be a little slow during the day.
- Overall it is comprehensive for an automated testing tool, but not that intuitive to use.



WAVE – Web accessibility tool

<http://www.wave.webaim.org/wave/index.jsp>

WAVE's accessibility report for a web page takes a very visual approach, with lots of icons on the screen highlighting accessibility issues. It may take a while to learn all the icons but a key is provided. You also have the option to install the WAVE check tool in your browser.



Part five: Testing the accessibility of your website

Cynthia

<http://www.contentquality.com/>

Cynthia is very much based on the format of the W3C Web Content Accessibility Guidelines Checklist as listed in part six.

You have a 'yes', 'no', and 'n/a' column for each checkpoint, with the tool providing comments on each checkpoint and giving line numbers of actual or possible errors which you need to check.



skip navigation | home | help | about this site | contact us | news | search | HiSoftware

Welcome to the HiSoftware® Cynthia Says™ Portal

The HiSoftware Company CynthiaSays portal is a joint Education and Outreach project of HiSoftware, ICDRI, and the Internet Society Disability and Special Needs Chapter. Read [Understanding Accessibility](#) today! [Download Now](#)

The HiSoftware Cynthia Says is a web content accessibility validation solution, it is designed to identify errors in your content related to Section 508 standards and/or the WCAG guidelines. Unlike HiSoftware's Desktop Software, [AccVerify](#), this online test only validates one page at a time.

Note this demo will test about one (1) page per minute / per site. This service will expose the [underlying technology and the benefits](#) of using the HiSoftware Solutions. [Buy HiSoftware AccVerify® today](#)

Testing with Adaptive Technology

There is a range of adaptive technology you can test your pages with, such as screen readers and magnifiers.

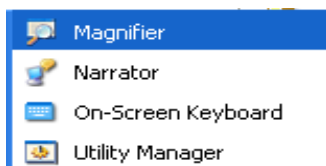
By far the most important adaptive technology to test with is a screen reader. There is a wide range of screen readers available today, some of the most popular are Hal, Jaws, Window-Eyes and a voice browser called IBM HomePage Reader.

All have their strengths and weaknesses and it can take time to learn how they work.

In the links section on the Resources CD you will find a list of demo and trial versions of the above screen readers you can download from the web.

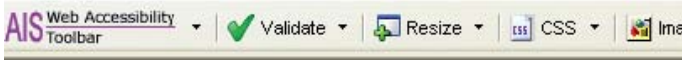
In addition on the Resources CD in the accessibility tools section you will find a guide to the main shortcut or hotkeys used in the Jaws screen reader.

There are also a wide range of magnifiers such as ZoomText, Supernova and iZoom all of which are listed on the Resources CD. There is also a Magnifier built into Windows called 'Magnifier' which you can find under Start > All programs > Accessories > Accessibility. > Magnifier.



Using the AIS Accessibility Toolbar

The AIS Toolbar is one of the best tools to test your web pages for accessibility. Details of how to obtain a copy can be found in the Accessibility tools section of the Resources CD.



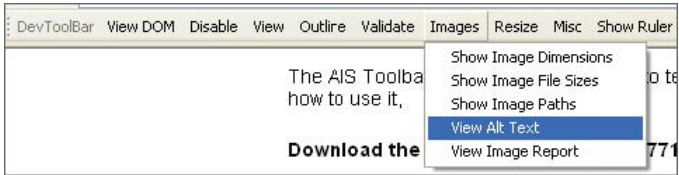
Testing a website for accessibility can be a time consuming and laborious process. Fortunately the free AIS Toolbar can do most of the hard work for you and is an indispensable tool for anyone interested in accessibility.

The toolbar is not an automated testing tool so does require manual work from you. It is therefore able to avoid the many problems with automated accessibility testing tools. It doesn't require any technical knowledge so even the biggest technophobe can check their website for accessibility!

There are 12 buttons in total on the toolbar, each with a down arrow to the right of the text. If you click on the down arrow for any of these buttons then a dropdown menu appears with all the available options. Alternatively you can use the keyboard shortcut keys assigned to each button.

Internet Explorer Developer Toolbar

An alternative to the AIS Toolbar is the Internet Explorer Developer Toolbar which is out in Beta and has many of the same accessibility testing features. More information about how to download a copy is included in the Accessibility tools section of the Resources CD.



Disabled user testing

In the UK disabled user testing has an increasingly important role in ensuring that a website is usable and accessible to disabled people.

This section gives an overview of how to find disabled testers, practical issues around testing and some of the benefits in this approach.

“We engaged AbilityNet to carry out disabled user testing on our corporate site and found this to be so beneficial that we intend to extend this testing to all of our websites...”

Adrian Barclay, Project Manager, Citizens Advice

Finding testers

One of the most challenging tasks in embarking on disabled user testing is finding representative people across a range of disabilities, covering the core groups of vision, hearing, mobility, cognitive and learning disabilities. It is important to find testers with a mixture of knowledge from expert to novice using a representative range of adaptive technology. Strategies include placing adverts on websites, mailing lists, special interest groups and contacting local and national disability groups.

As a VCS organisation you might not have a budget to do disabled user testing, in which case you need to focus on finding individuals you know in your local community who would be willing to help you out.

The typical range of disabled users in testing

- User with severe vision impairment - screen reader (such as Jaws, Hal and Supernova).
- User with moderate visual impairment - magnification software (such as ZoomText and Lunar).
- User with mild vision impairment - large text plus white on black colour preferences
- User with severe motor difficulties - switch/head operated mouse access and on-screen keyboard.
- User with medium motor difficulties/ULD - for example voice recognition or joystick user.
- User with mild motor difficulties - fine mouse control difficulties.
- User with moderate Dyslexia - changing colours and text to speech software (using software such as TextHelp/BrowseAloud).
- BSL (British Sign Language) user
- User with mild learning/cognitive difficulties - no special access method.

Practical considerations

Having found suitable candidates you need to consider how you are going to ensure successful user testing within the constraints of a limited time frame and often limited budget. Ensuring disabled testers do a series of tasks, obtaining their feedback and compiling the information into a report that is clear, informative and takes accounts of individual differences can be a challenge.

The two approaches you have are:

- Home testing.
- Moderated testing in the home or in your office.

A key question is how many testers should you use? This is not an easy to answer given you must balance cost against the need to have a representative sample of testers. The main problem is the fewer the number of testers the more likely that there could be erroneous findings due to the results of one individual.

Other things that need to be considered are:

- Capturing the software and hardware profile of each disabled individual to ensure you can have the necessary software and hardware in place for moderated testing in your offices.
- Ensuring user tasks are not biased against one disability group or another because of the language used or the nature of the task.
- If a moderator is used make sure they are sensitive to the needs of disabled testers and have knowledge of adaptive technology.

Key benefits of disabled user testing

The benefits of disabled user testing include:

- A deeper understanding of accessibility issues from a users perspective.
- A practical insight into the importance of site usability for disabled people.
- Disabled user testing complements W3C accessibility checkpoint audits – giving you a more comprehensive accessibility evaluation of your site.

Testing for specific accessibility issues

This section takes you through point by point how to test for a range of the most common accessibility issues. At the end of the section is a checklist you can use to help you take a systematic approach to auditing your site. You will also find a Word copy in the Accessibility tools section of the Resources CD.

How to test for accessible alt text

Tools to use: AIS Toolbar, Internet Explorer, Lynx, screen reader.

Instructions

In the AIS Toolbar click on the dropdown arrow next to the 'Images' menu and select 'Show Images'. On the page you will see alt tags for those graphics that have them - check they are descriptive and make sense. For images that are missing alt text you will see a red **No Alt!** next to them. You can also try out the other options under the 'Images' menu for interest.

How to test scripting & programmatic elements

Tools to use: AIS Toolbar, Firefox, Opera, screen reader.

Instructions

Identifying scripts & programmatic elements on a page:

In the AIS Toolbar click on the dropdown arrow next to the 'Structure' menu and select 'Event Handlers'.

You will see a summary box telling you how many event handlers are on the page. If there are any, you will see icons on the page where they are located. The one to look out for is the onmouseover attribute as it has the most accessibility problems.

Testing for Flash accessibility:

1. In the AIS Toolbar click on the dropdown arrow next to the 'Tools' menu then select 'Simulations' then 'Disable Plug-ins'. The Flash will disappear from the page, check if you can still navigate around the page and if the page is understandable.
2. Using just the keyboard tab through the page - can you tab through the Flash movie and highlight and select any of the links in the movie?
3. Run a screen reader such as Jaws through the page, does it pick up information about the Flash movie? With Jaws running test the Flash links work by pressing 'Insert' + 'F5' this will bring up any Flash buttons on the page as well as any form fields. Look for unlabelled 'Button' links which indicates the Flash movie is not fully accessible.

JavaScript functionality

In the Firefox browser go to the 'Tools' menu and select 'Options'. Under 'Options' select 'Web features' and untick the box next to 'Enable JavaScript', then click OK.

Test that all the page features work now that you have turned off JavaScript. Particularly look at form submission and your site search.

How to test Multimedia

Tools to use: n/a

Instructions

For multimedia embedded in the page check if captions or transcripts are provided.

If the media file launches in a stand alone media player such as Windows Media Player, turn on the captions option via the play menu - can you see captions at the bottom of the movie? Alternatively has a transcript been provided?

Note: In Windows Media Player 10 you might need to change your security settings to view captions. Go to the 'Tools' menu and select 'Options'. Click on the 'Security' tab and make sure the box next to 'Show local captions when present' is ticked, then click 'OK' to return to the media player.

How to test Cascading Style Sheets (CSS)

Tools to use: AIS Toolbar, IE, Opera

Instructions

In the AIS Toolbar click on the dropdown arrow next to the 'IE Options' menu and select 'Toggle CSS'. After a moment the page will reload with style sheets turned off, look at the page and check that all the text and information is still readable.

Alternatively if you know how switch off style sheets using your browser you can do that to test the page. For example in Opera 8 you can turn off style sheets under the 'Advanced' tab options section of 'Preferences' (The keyboard shortcut is 'Ctrl' + 'F12').

How to test data tables

Tools to use: AIS Toolbar

Instructions

In the AIS Toolbar click on the dropdown arrow next to the 'Structure' menu and select the 'Simple Data Table' option. All the table cells will be highlighted on the page. Check if column headings and rows (if present) are labelled correctly with the <th> tag and the scope attribute has been used.

For more complex tables with multiple levels of data repeat the above instructions selecting 'Complex Data Table' instead of 'Simple Data Table'.

How to test Frames and iframes

Tools to use: AIS Toolbar

Instructions

In the AIS Toolbar click on the dropdown arrow next to the 'Structure' menu and select 'Frame Name /Title'.

Frame names and titles will be listed in a new window. Check that meaningful titles have been used for each frame.

How to test using colour to convey information/ colour contrast

Tools to use: AIS Toolbar, Colour contrast analyser tool, laptop and or older monitor.

Instructions

1. In the AIS Toolbar click on the dropdown arrow next to the 'Colours' menu and select 'Greyscale'. Look at the page for any contrast issues.
2. Use the AIS Colour contrast Analyser tool select foreground and background colours from the page using the 'pick colour' icons and test for contrast levels.
3. Test on a low spec monitor and laptop screen.

How to test accessible link text

Tools to use: AIS Toolbar, Opera, screen reader

Instructions

In the AIS Toolbar click on the dropdown arrow next to the 'Doc Info' menu and select the 'List Links' option. Read through the links and check they make sense.

You can also use a screen reader to check for clear hypertext links, for example in Jaws you can generate a links list by pressing 'Ctrl' + 'F7'.

Alternatively you can use Opera 8 to list links on a page - from the 'Tools' menu select 'Links'.

How to test relative sizing

Tools to use: IE , AIS Toolbar, PDA

Instructions

Resize text (Tool – IE)

In Internet Explorer (IE) from the 'View' menu select 'Text Size' and change to the 'Larger' or 'Largest' font setting. If the font size does not change then the font has been fixed.

Resizing browser (Tool – AIS Toolbar)

In the AIS Toolbar click on the dropdown arrow next to the 'Resize' menu and change the resolution to 800 x 600 - does the page resize well?

Also test the web page on a PDA – is it legible?

How to test for accessible forms

Tools to use: AIS Toolbar

Instructions

In the AIS Toolbar click on the dropdown arrow next to the 'Structure' menu and select 'Fieldset/Labels'.

Fieldset and label elements will be highlighted on the page. Make sure a 'for' attribute and an 'id' attribute have both been used for each form field.

How to test header structure

(Tools - AIS Toolbar)

Instructions

In the AIS Toolbar click on the dropdown arrow next to the 'Structure' menu and select 'Headings'. If used, header tags will be displayed for example `<h1></h1>` wrapped around headings on the page.

Testing checklist

Alternative ways of viewing your content:

- Do all pictures/images have relevant descriptive alt text?
- Have you used pictures and symbols in addition to text if appropriate?
- Are your pages usable with scripts/applets turned off?
- Are your pages readable with style sheets (CSS) turned off?
- Have you provided at least a text equivalent for each non-text element, for example Flash and Java applets? Ideally provide an html equivalent.

Links:

- Does each text contain information about its destination and is it unique? For example don't use 'click here' or 'more'.
- Do links that open in a new window have a clear warning?
- Do links have sufficient space between them and are they big enough to be easily clickable?
- Has title text been added to text links to give more information if necessary?

Language/style:

- Have you used the simplest and clearest language possible for your site?
- Have you used short line lengths and paragraphs and clear descriptive headings?
- Have you indicated changes in language (for example English to French) if you use more one language on a page?

Tables and frames

- For data tables have you coded row and column headings with the <th> tag?
- If you have used frames, have you added clear descriptive titles to each one?

Colour:

- Have you avoided using colour alone to convey #information?
- Is there sufficient contrast on text images and between text and background colours?
- Have you checked your site colour scheme is colour blind friendly?

Structure/layout:

- Have you provided consistent navigation throughout the site?
- Have you clearly defined the different sections of the page and ensured consistent location of screen objects such as your logo, search box and navigation bars?
- Do your pages have a logical tab order through all screen links, forms and Flash movies?
- Does the page text resize in Internet Explorer?
- Are your pages still legible when you increase the text size – no cropping or overlapping?
- Is the majority of your text left justified?

Moving images:

- Do you avoid distracting continuous movement on your web pages?

Mark-up

- Have you checked your page code is valid (W3C html validator)?
- Have you checked your style sheets validate (W3C CSS validator)?

Forms

- For all form fields has the <label> tag been used to mark up its description?
- For each form field does the label 'for' value match the input fields 'id' value?
- For checkboxes and radio buttons is the label on the right hand side?

Acrobat PDF Files

- Do your PDF files pass the accessibility quick check tool in Adobe reader software?
- Have the PDFs been tagged properly? Are bookmarks present in the left hand frame?

Part six:

Resources

Glossary

Accessibility (web) - Designing sites so as many people as possible can access and interact with them effectively and easily, independent of who they are or how they access the web.

Alternative text (alt text) - Descriptive text attached to any non-text element such as images, movies and animations on a web page written in html. It is particularly important for blind web users who use assistive technology called screen readers to make sense of a page.

Assistive/Adaptive technology - Computer software or hardware that makes it possible for people with disabilities to access computer systems. Examples include screen readers and magnifiers, closed captioning, alternative keyboards and mice.

Cascading Style Sheet (CSS) - a separate file linked to a web page that contains the rules for how the page should look in terms of colours, font styles and size and layout.

Content Management System (CMS) -

Software that enables the controlled update of the content of websites, where many pages of content are published and/or many people are involved in the work flow processes needed to create and publish that content.

Disability Discrimination Act (DDA) - the legislation in the UK that covers the rights of disabled people. The section of the act on the provision of goods and services includes the requirement that websites should be made accessible.

Flash - Multimedia technology developed by Adobe (formerly Macromedia) used for web animation and often used to build websites with rich dynamic content. Historically it has not been very compliant with adaptive technology but with each new version the accessibility features improve.

Frames - A feature of html that allows a page to be divided into two or more separate windows. If the frame does not have a <title> element, or the <title> element is not meaningful this can cause accessibility issues.

HyperText Markup Language (html) - A common markup language used to create web pages.

JavaScript - A scripting language commonly used on web pages. It has many uses, including validating fields in a form, or writing information to the user's screen. Adaptive technology such as screen readers don't always support JavaScript which why its use is an accessibility issue.

Lynx - A text only browser that is popular with people with disabilities and those in low bandwidth areas.

Screen reader - Software that reads the content of a computer screen aloud. Screen readers can only interpret text content, so all graphic and multimedia must have alternative text descriptions using alt text, captions, transcripts, or other methods.

Section 508 - This is a common name for Section 508 of the Rehabilitation Act. This is an amendment to a US law that basically says all Electronic and Information Technology purchased or developed by the US Government must be accessible to people with disabilities.

Spacer Images - also called spacer gifs. These are small transparent images placed on a page, usually in a table used for layout. They help to place text and images on the page for a good visual effect.

World Wide Web Consortium (W3C) - an international consortium of companies and organisations involved with the Internet and the web. They are responsible for maintaining web technology standards, such as html and CSS.

Web Accessibility Initiative (WAI) - started by W3C and its members, it addresses web accessibility issues for web content, browsers and developer tools used to build websites.

Web Content Accessibility Guidelines (WCAG) - These are the guidelines created by the W3C/WAI to address issues in building accessible web pages.

Extensible Hypertext Markup Language (xhtml) - This is an updated version of html, which uses more rigorous standards and rules to make better structured and accessible web pages.

Validation - Each web page should conform to a specified standard (document type) which is normally placed in the web page code. If when checked with a validation tool the page passes the standard it said to be valid. The benefit of valid pages is that they work more reliably with web browsers and a range of adaptive technology used by disabled people.

Useful websites

Below are a selection of useful websites on web accessibility; all of them are listed in the link section of the Resources CD that comes with this guide.

Accessibility guidelines resources

Web Accessibility Initiative (WAI)

<http://www.w3.org/WAI/>

Checklist of Checkpoints for Web Content Accessibility Guidelines 1.0

<http://www.w3.org/TR/WCAG10/full-checklist.html>

Overview of Web Content Accessibility Guidelines (WCAG) 2.0 (draft)

<http://www.w3.org/WAI/intro/wcag20.php>

HTML Techniques for Web Content Accessibility Guidelines 1.0

<http://www.w3.org/TR/WCAG10-HTML-TECHS/>

Section 508

<http://www.section508.gov/>

Websites on accessibility

My Web My Way

My Web My Way contains lots of useful information on how to change your computer and browser settings to make the web easier to use.

<http://www.bbc.co.uk/accessibility/>

Webaim

Webaim.org is one of the best accessibility resources on the web.

<http://www.webaim.org/>

A List Apart

A List apart is a resource of articles covering web standards including accessibility and usability.

<http://www.alistapart.com/>

CSS Zen Garden

The cutting edge of CSS design – moving away from table design.

<http://www.csszengarden.com/>

RNIB's Web Access centre

http://www.rnib.org.uk/xpedio/groups/public/documents/publicwebsite/public_webaccessibility.hcsp

W3C Web Content Accessibility Guidelines Checklist levels 1 and 2

This checklist is a summary of the **Web Content Accessibility Guidelines (WCAG)**. Each numbered checkpoint below corresponds to a section in the full **Web Content Accessibility Guidelines** which can be found at: <http://www.w3.org/WAI/>

Priority 1 checkpoints

In General (Priority 1)

1.1 Provide a text equivalent for every non-text element (e.g., via “alt”, “longdesc”, or in element content). This includes: images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ascii art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video.

2.1 Ensure that all information conveyed with colour is also available without colour, for example from context or markup.

4.1 Clearly identify changes in the natural language of a document’s text and any text equivalents (e.g., captions).

6.1 Organize documents so they may be read without style sheets. For example, when an html document is rendered without associated style sheets, it must still be possible to read the document.

6.2 Ensure that equivalents for dynamic content are updated when the dynamic content changes.

7.1 Until user agents allow users to control flickering, avoid causing the screen to flicker.

14.1 Use the clearest and simplest language appropriate for a site’s content.

And if you use images and image maps

1.2 Provide redundant text links for each active region of a server-side image map.

9.1 Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

And if you use tables

5.1 For data tables, identify row and column headers.

5.2 For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells.

And if you use frames

12.1 Title each frame to facilitate frame identification and navigation.

And if you use applets and scripts

6.3 Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.

And if you use multimedia

1.3 Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation.

1.4 For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation.

And if all else fails

11.4 If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page.

Priority 2 checkpoints

In General

2.2 Ensure that foreground and background colour combinations provide sufficient contrast when viewed by someone with colour blindness or when viewed on a black and white screen. [Priority 2 for images, Priority 3 for text].

3.1 When an appropriate markup language exists, use markup rather than images to convey information.

3.2 Create documents that validate to published formal grammars.

3.3 Use style sheets to control layout and presentation.

3.4 Use relative rather than absolute units in markup language attribute values and style sheet property values.

3.5 Use header elements to convey document structure and use them according to specification.

3.6 Mark up lists and list items properly.

3.7 Mark up quotations. Do not use quotation markup for formatting effects such as indentation.

6.5 Ensure that dynamic content is accessible or provide an alternative presentation or page.

7.2 Until user agents allow users to control blinking, avoid causing content to blink (i.e., change presentation at a regular rate, such as turning on and off).

7.4 Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages.

7.5 Until user agents provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects.

10.1 Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user.

11.1 Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported.

11.2 Avoid deprecated features of W3C technologies.

12.3 Divide large blocks of information into more manageable groups where natural and appropriate.

13.1 Clearly identify the target of each link.

13.2 Provide metadata to add semantic information to pages and sites.

13.3 Provide information about the general layout of a site (e.g., a site map or table of contents)

13.4 Use navigation mechanisms in a consistent manner.

And if you use tables

5.3 Do not use tables for layout unless the table makes sense when linearized.

Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version).

5.4 If a table is used for layout, do not use any structural markup for the purpose of visual formatting.

And if you use frames

12.2 Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone.

And if you use forms

10.2 Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned.

12.4 Associate labels explicitly with their controls.

And if you use applets and scripts

6.4 For scripts and applets, ensure that event handlers are input device-independent.

7.3 Until user agents allow users to freeze moving content, avoid movement in pages.

8.1 Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies [Priority 1 if functionality is important and not presented elsewhere, otherwise Priority 2.]

9.2 Ensure that any element that has its own interface can be operated in a device-independent manner.

9.3 For scripts, specify logical event handlers rather than device-dependent event handlers.

Helpful organisations

ICT Hub Partner organisations

AbilityNet

Address: PO Box 94

Warwick

Warwickshire CV34 5WS

Telephone: 0800 269545

Fax:01926 407425

Website:<http://www.abilitynet.org.uk>

Email:enquiries@abilitynet.org.uk

AbilityNet supports anyone with a disability to access ICT, sets standards in accessible web design and trains 4,000 professionals a year in healthy computer use and the benefits of accessible ICT for all. It is committed to ensuring everyone who wishes can benefit from the equality of opportunity that using a computer brings.

IT4 Communities

Address: CentralPoint

45 Beech Street

London EC2Y 8AD

Telephone: 020 7796 2144

Fax:01926 407425

Website: <http://www.it4communities.org.uk>

Email: info@it4communities.org.uk

IT4 Communities provides volunteer IT support to the voluntary and community sector through a network of IT professionals. Launched in November 2002, it has delivered over £580,000 of support to date with a return on investment of over 3:1.

LASA - London Advice Services Alliance

Address: Universal House

88-94 Wentworth Street

London E1 7SA

Telephone: 020 7377 2748

Fax:020 7247 4725

Website: <http://www.lasa.org.uk>

Email: info@lasa.org.uk

LASA offers comprehensive online ICT information for small and medium sized organisations, written in plain English, for non-specialists. It also hosts the UK Circuit Rider steering group. Circuit Riders are IT professionals offering mobile support to local organisations.

NACVS – National Association of Councils for Voluntary Service

Address: 177 Arundel Street
Sheffield S1 2NU

Telephone: 0114 278 6636

Textphone: 0114 278 7025

Fax: 0114 278 7004

Website: <http://www.nacvs.org.uk>

Email: nacvs@nacvs.org.uk

NACVS is a network of over 300 Councils for Voluntary Service providing local support and a national voice for these organisations. It hosts an IT email list with over 130 members, and provides IT services, training and events for its members.

NCVO – The National Council for Voluntary Organisations.

Address: Regents Wharf

8 All Saints Street

London N1 9RL

Telephone: 020 7713 6161

Fax: 020 7713 6300

Helpdesk: 0800 2798 798

Textphone: 0800 0188 111

Website: <http://ncvo-vol.org.uk>

Email: ncvo@ncvo-vol.org.uk

NCVO has over 10 years experience of increasing the capacity of the voluntary and community sector through ICT. It organises events, lobbies and works with funders, hosts a forum of over 500 ICT practitioners, and has negotiated discounts with suppliers saving the sector in excess of £10million over 5 years. It also operates a free ICT advice and information helpdesk.