

Activedition & Accessibility

Introduction

This paper looks at why web site accessibility has become such an important consideration for new websites and intranets. As well as giving a general overview on accessibility and Activedition it looks at the W3C accessibility guidelines v1 in detail and how Activedition is designed to assist with compliance. We believe that Activedition is the leading CMS for accessibility compliance delivery and this is backed by independent accessibility experts. We are also committed to the ongoing development of new features to assist users in delivering best practice websites.

Why Web Site Accessibility?

Web site accessibility has become a major Internet topic in recent years. Thanks to innovative browsers and specialised hardware disabled users can now actively access Internet services. One example of an assistive technology is a browser based 'screen reader' which reads the site out aloud to the end user. However, the site needs to be designed and implemented in a way that will allow these kinds of assistive technologies to operate effectively. Some of the main reasons why leading organisations are building accessibility into their web sites are:

1. Legal Compliance

Under the UK Disability Discrimination Act, services that are provided online must be made accessible to disabled users. Any legal action based on the Disability Discrimination Act will have a significant legal and PR cost.

2. Appealing to a Wider Market

According to the Office for National Statistics, there is approximately 8.5 million disabled customers in the UK. An inaccessible site shuts out many of these people as potential customers.

3. Social Responsibility

It's simply the right thing to do. With a little effort and no additional cost in the design and implementation stage, organisations can improve the lives of those with disabilities. There are also the potential PR benefits that such an approach will provide.

Up until recently most web sites were not designed with accessibility in mind. However, more organisations are beginning to realise the sound commercial benefits of making their sites both accessible and usable. One of the myths concerning accessibility is that sites that are accessible will have no visual appeal. Recent efforts by some leading organisations are showing that there does not need to be a trade-off between aesthetic design and accessibility.

Activedition & Accessibility

Accessibility & Activedition

Overview

Activedition is an Enterprise Web Content Management solution that has a wide range of accessibility features and functions built in as standard. The accessibility features that Activedition provides are typically optional for the end user when creating content. However, a number of features can be configured through Activedition's system options so that good accessibility practices are enforced at the content creation and update stage. Activedition currently supports the WAI's priority 1 - 3 and the RNIB's "See It Right" accessibility guidelines for content. The diagram below show the accessibility system options available with Activedition.

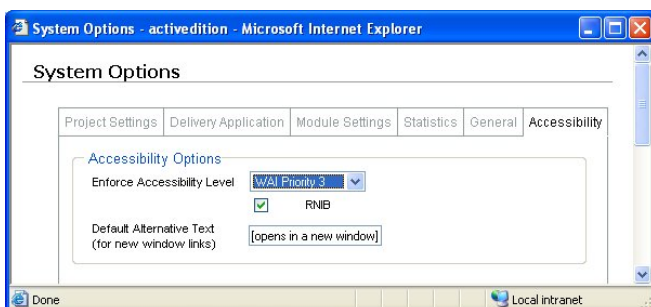


Figure 1 - System accessibility option with Activedition

There are 3 elements which make up an effective accessibility solution for a content managed website.

1. **Template Design** - templates must be designed to the chosen accessibility level, A, AA or AAA. For example the template design must ensure that foreground and background colour combinations provide sufficient contrast when viewed by someone having colour deficits or when viewed on a black and white screen.

2. **User Education** - there are some simple best practices that content contributors require to be aware of when drafting content e.g. do not use a

link with a term "click here" as this could confusing to a blind person using a screen reader.

3. **Content Management Conformance and Enforcement** - the content management system also needs to provide for features to allow the interactive functions to conform to an accessible environment. Through the Activedition accessibility enforcement switch the CMS can assist with ongoing accessibility conformance. There are a number of accessibility focused features within Activedition that facilitate ongoing publishing to W3C & RNIB standards:

- Enable A, AA & AAA Compliant Delivery
- Enable RNIB Compliant Delivery
- Accessibility Friendly WYSIWYG Editor
- Accessible Multimedia Support - ALT & Long Description
- Accessible Forms - Labels, Tab Order, Default Values, Client & Server Side Checking
- No Reliance on Assistive Technologies e.g. Javascript
- Liquid Layout & Relative Sizing
- Accessible Tables - Summary, Row & Column Header Support
- Link Access Keys & Title Support
- Accessibility Enforcement Options
- Delivery to both XHTML & HTML 4 Standards
- Automated Text Only Version Generation
- Accessible Navigation Components e.g. Breadcrumb trail
- Template Accessibility Checks
- Spell Checker on Site Search
- Accessible Tools e.g. Site Map, Site Search, Discussion Forums, What's New, Newsletters
- Separation of Content from Presentation - CSS driven for presentation

Activedition & Accessibility

W3C Accessibility Guidelines in Detail

Priority 1 Checkpoints

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.

ALT & longdesc attributes are supported in the multimedia library allowing graphical elements to have default accessibility information to be defined. These are then automatically placed whenever the multimedia is used. The default values can be overridden for each specific placement.

These can be made mandatory by turning on the accessibility option in system options.

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

No elements generated by Activedition or supporting custom features rely on colour to convey information.

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

The WYSIWYG editor within Activedition supports the use of the SPAN tag allowing business users to embed pieces of content in other languages. Non-technical users can access this facility without the need to access the HTML text editor. The system also supports pages in all languages with full UTF-8 support.

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.

All HTML code generated by Activedition is designed to be read without stylesheets.

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

All dynamic elements generated by Activedition apply this guideline e.g. Search results, Site Map etc.

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

No Activedition generated code elements will cause the screen to flicker.

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

The content generated by Activedition looks to ensure that the supporting content is clear, simple and instructive.

And if you use images and image maps (Priority 1)
1.2 Provide redundant text links for each active region of a server-side image map.

Activedition does not generate any image maps.

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.

Activedition does not generate any image maps.

And if you use tables (Priority 1)
5.1 For data tables, identify row and column headers.

Activedition applies these rules for all data tables it produces e.g. Newsgroups. The WYSIWYG editor within Activedition also supports the creation of row

Activedition & Accessibility

and column headers, ensuring that users can create and update accessible content.

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.

Activedition supports this on all generated data tables and automatically associates data and header cells.

And if you use frames (Priority 1)

12.1 Title each frame to facilitate frame identification and navigation.

Activedition does not generate framed pages.

And if you use applets and scripts (Priority 1)

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.

All Activedition generated code is designed to operate with scripting support disabled. The tag system allows administrators to embed applets etc., but these must be designed to meet this requirement.

And if you use multimedia (Priority 1)

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.

No code generated by Activedition is affected by this requirement. All information uploaded to the multimedia library in Activedition should consider this requirement.

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.

No code generated by Activedition is affected by this requirement. All information uploaded to the multimedia library in Activedition should consider this requirement.

And if all else fails (Priority 1)

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.

The Activedition content management features allows the contributors to achieve this with ease.

Priority 2 Checkpoints

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

The content management facilities are designed to make it simple for non-technical user to create content in adherence with this checkpoint.

All templates and stylesheets designed for the site should also consider this requirement.

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

Activedition is designed to utilise markup rather than images to convey information.

3.2 Create documents that validate to published formal grammars.

Activedition generates code that validates to HTML 4.01 Transitional and XHTML 1.0 Transitional. We anticipate supporting the Strict variations of these grammars in the future.

3.3 Use style sheets to control layout and presentation.

Activedition utilises templates and cascading stylesheets to control layout and presentation. As much layout and presentation information as desired can be placed in the CSS as desired by the site designers.

Activedition & Accessibility

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

All code generated by Activedition utilises relative sizes.

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

Header elements are fully supported in the WYSISYG editor. This is further complemented with the separation of the page header from the supporting text on the page. Finally sections help to split up the page and encourage the use of section headers.

3.6 Mark up lists and list items properly.

Correct list item markup is fully supported in the WYSIWYG editor.

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

Quotations are supported in the WYSIWYG editor. The code generated by Activedition does not utilise this markup for indentation or other formatting effects.

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

All content produced by Activedition is designed to be accessible, including the dynamic elements. An alternative text-only presentation is also automatically supported.

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).

No standard Activedition features utilise blinking or flashing content.

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

Activedition does not generate any pages that periodically auto-refresh. A check in the template parser will warn users of this checkpoint if they utilise a Meta refresh tag.

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.

Activedition does not utilise any client side redirects in the generated code.

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.

With the accessibility enforcement feature enabled, all links that open in a new window are prefixed with a configurable string of text e.g. Opens in a new window. This assists users in conforming with this checkpoint.

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

Activedition currently generates code to validate against HTML 4.01 Transitional and XHTML 1.0 Transitional. These are the current W3C standards for HTML. In coming releases we also anticipate supporting the STRICT versions of these grammars. W3C standards for CSS are fully supported.

11.2 Avoid deprecated features of W3C technologies.

All deprecated elements and attributes have been removed from Activedition generated code.

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

The content management facilities are designed to make it simple for non-technical user to create content in adherence with this checkpoint.

13.1 Clearly identify the target of each link.

Activedition & Accessibility

Support for text on links that open a new window. This is further enhanced with title support on links.

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

The extensive Meta data system allows Meta-data to be defined and supported to specific customer requirements.

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

Activedition automatically generates and maintains a site map as standard.

The site map has been reworked from previous releases to provide a more user friendly and accessible format.

13.4 Use navigation mechanisms in a consistent manner.

Configurable user navigation components can be embedded in templates to provide a consistent navigation mechanism. An automated breadcrumb trail is also supported to illustrate to the user where they are in the site structure. Markup of navigation has been improved to define a hierarchical structure.

And if you use tables (Priority 2)

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).

Where Activedition generates table for layout they are designed to make sense when linearized

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

No structural markup is used for the purpose of visual formatting.

And if you use frames (Priority 2)

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

Activedition does not use frames in the generated code.

And if you use forms (Priority 2)

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.

Where appropriate labels are used and are properly positioned.

12.4 Associate labels explicitly with their controls.

For all forms (generated and standard) labels are associated explicitly with their controls.

And if you use applets and scripts (Priority 2)

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

Activedition generated code is designed to operate with script support disabled. Client side validation is complemented with server side validation checks.

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

No standard Activedition code makes use of 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.]

Activedition functionality is designed to work with assistive technologies, but it does not rely on their existence.

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

Activedition & Accessibility

Activedition code generates valid HTML/XHTML that can be operated in a device-independent manner.

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

Activedition scripts are designed to look at actions rather than device-dependent events e.g. onsubmit rather than onclick.

Priority 3 Checkpoints

4.2 Specify the expansion of each abbreviation or acronym in a document where it first occurs.

Acronym support is standard in the WYSIWYG editor. The expansion of each acronym on its first use in the document is the responsibility of the content editor.

4.3 Identify the primary natural language of a document.

This is supported via the templates that are uploaded to the site.

9.4 Create a logical tab order through links, form controls, and objects.

All generated code has the appropriate tab order automatically generated.

9.5 Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls.

Keyboard shortcuts are supported on all links via the Link Library and on all placement of links.

10.5 Until user agents (including assistive technologies) render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links.

Adherence to this checkpoint is the responsibility of the template author and content contributors.

11.3 Provide information so that users may receive documents according to their preferences (e.g., language, content type, etc.)

Adherence to this checkpoint is the responsibility of the template author and content contributors.

13.5 Provide navigation bars to highlight and give access to the navigation mechanism.

Configurable text based navigation and an automated breadcrumb trail look to ensure that the user can easily access the navigation mechanism.

13.6 Group related links, identify the group (for user agents), and, until user agents do so, provide a way to bypass the group.

The Activedition section system provides the ideal framework for grouping pieces of content. The ability to identify and bypass these can be built into the page templates that are uploaded for use on the site.

13.7 If search functions are provided, enable different types of searches for different skill levels and preferences.

Activedition provides a simple site search facility and an advanced search facility as standard. This can be further complemented with customer specific Meta data based search facilities.

The spell checking facilities have also been integrated into the site search to make suggestions on words that the user may have misspelled e.g. Did you mean "head" instead on "heas"

13.8 Place distinguishing information at the beginning of headings, paragraphs, lists, etc.

The content management facilities are designed to make it simple for non-technical user to create content in adherence with this checkpoint.

13.9 Provide information about document collections (i.e., documents comprising multiple pages.).

Activedition & Accessibility

The content management facilities are designed to make it simple for non-technical user to create content in adherence with this checkpoint.

13.10 Provide a means to skip over multi-line ASCII art.

HTML Templates and HTML editing support allow experienced HTML users to create anchors to skip any content including ASCII art.

14.2 Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page.

The content management facilities are designed to make it simple for non-technical user to create content in adherence with this checkpoint.

14.3 Create a style of presentation that is consistent across pages.

Templates and stylesheets within Activedition are designed to ensure conformance with this requirement

And if you use images and image maps (Priority 3)
1.5 Until user agents render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map.
Activedition does not make use of client-side image maps.

And if you use tables (Priority 3)
5.5 Provide summaries for tables.

Table summaries are supported in the WYSIWYG editor. All Activedition supporting features that use tables implement a table summary e.g. newsgroups.

5.6 Provide abbreviations for header labels.

This is supported in the tables generated by Activedition.

10.3 Until user agents (including assistive technologies) render side-by-side text correctly, provide a linear text alternative (on the current page or some other) for all tables that lay out text in parallel, word-wrapped columns.

A text only version of all pages is provided in Activedition and text sections can be laid out differently in the text only alternative of the page (e.g. linear fashion)

And if you use forms (Priority 3)
10.4 Until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas.

Default values can be specified in all form fields ensuring compliance with this checkpoint.
Implementation

Key to delivering an accessible site on Activedition is the implementation. Whilst Activedition has a range of features that make delivering an accessible website possible it does not and cannot prevent accessibility mistakes. Specifically the implementation of any site must consider:

- Page Templates
- Stylesheets
- Custom Code Pages
- Tags (code snippets uploaded to the Tag Manager)
- Content Population, Maintenance & User Training
- Content Updates & Reviews
- Conformance Testing

Consideration of these elements in combination with the Activedition CMS will ensure both the initial and ongoing compliance with accessibility guidelines.

Conclusions

The power of the web is in its universality. Access for all is a vital aspect and there are now accessibility guidelines in place that offer best practice advice for the accessible design and implementation of websites. The Activedition Enterprise Web Content Management System provides a range of out-of-the-box features which can assist in delivering and maintaining websites that adhere to the current accessibility guidelines.