Education Services Australia

ESA guidelines for online content

Version 8.0

17 October 2014

Disclaimer

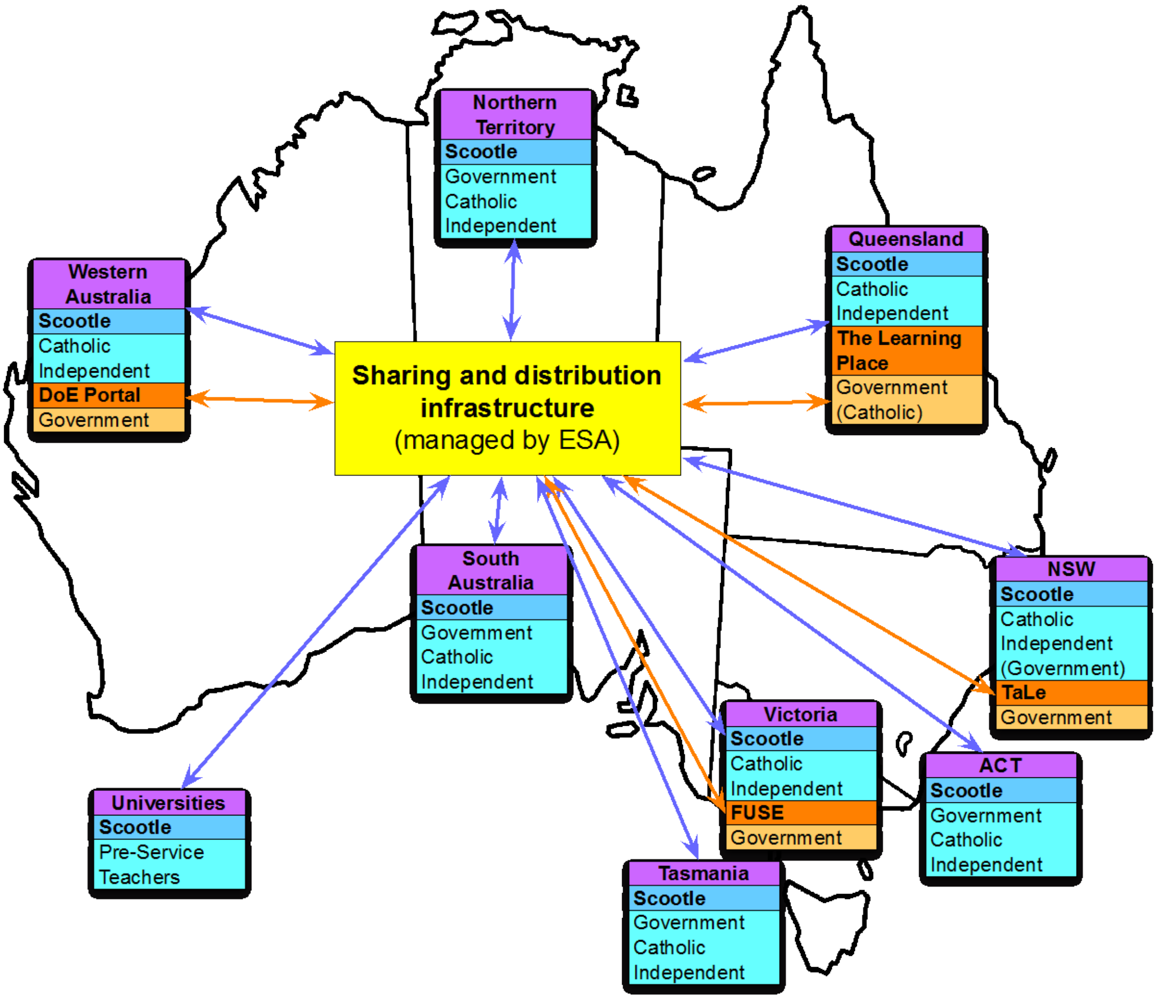
The material contained in the *ESA guidelines for online content* is for general information purposes only. Any use of the material is at your own risk. To the extent permitted by law, Education Services Australia will not be liable for any loss or damage suffered as a result of any party relying upon the information contained in the document *ESA guidelines for online content*.

Contents

1.0 Introduction

Education Services Australia (ESA) has developed the *Guidelines for online content* to provide information on requirements for developing effective websites, applications and resources for use in the Australian education sector. It is also of assistance to web developers, commercial publishers, system administrators, schools and teachers who want to share their resources via the National Digital Learning Resources Network and other repositories managed by the Australian education jurisdictions.

ESA manages the National Digital Learning Resources Network (NDLRN) to provide sharing and distribution infrastructure available to all jurisdictions. ESA also manages the Scootle website, which provides a search and access tool for the resources in NDLRN. Content in the NDLRN is available to schools either through a secure connection to Scootle or via their own local network. Jurisdictions can download content from the NDLRN to their own server or upload and share content within their own schools or any of the other jurisdictions. The use of the network by the jurisdictions is shown below.



Two distribution models are provided for education jurisdictions:

- digital curriculum resources and metadata are supplied from a secured network
- delivery of the metadata only, directing users to partner websites to access individual digital curriculum resources

There are four core systems that make up this infrastructure:

- the Learning Exchange (LEX), a framework for creating, distributing and managing digital curriculum resources and intellectual property rights. This is a legacy system and no new content is loaded.
- the Sharing Exchange (SHEX), a repository for the Australian school sector that enables digital resources to be shared. This is the primary storage for all new content.
- the Metadata Exchange (MEX), a system that harvests the metadata records of digital resources suitable for schools in Australia from a range of content creators such as museums, non-government organisations and commercial publishers.
- the Publishers Exchange (PREX), a system that harvests the metadata records of digital resources suitable for schools in Australia from commercial publishers.

Purpose

The *Guidelines for online content* document provides information to assist web developers in identifying requirements and practices that will enable them to create content that is dynamic, robust and usable over all education jurisdictions. It is intended as a guide rather than a prescriptive listing of standards and specifications because the web is changing so quickly and the practices used are evolving as the available functionality expands.

The guidelines cover websites, online apps, interactives and multimedia. ESA uses these guidelines in making decisions on technology to be used in developing online content, and works closely with jurisdictions to ensure that the functionality is compatible with their requirements.

Throughout the document the requirement for the development of dynamic and accessible products operating for the widest possible range of users is emphasised. The range of devices used to access online content is expanding rapidly, and the dependence on information provided online makes it imperative that these considerations are taken into consideration at the outset of all online development projects.

Obligations of Education Services Australia

These guidelines contribute to national and international standards development for online content through an evolving body of knowledge and practice. Education Services Australia will fulfil the following obligations in managing these guidelines by:

- consulting with relevant education jurisdictions, organisations, user groups and developers in distributing, using and reviewing these guidelines
- reviewing and evolving the guidelines to reflect knowledge and practice derived from the development, deployment and use of online content

Development, monitoring and usage

The *Guidelines for online content* have evolved from technical specifications developed by The Learning Federation (TLF) project from 2003 to 2009. The National Digital Learning Resources Network (NDLRN) now manages the learning objects and digital resources created by TLF. This document reflects the changes in requirements as the number of digital learning resources created inside of school authorities is reduced, and development shifts to a range of online content created for delivery on a wide range of devices.

As a result, this document no longer prescribes minimum technology requirements or standards for compliance, as previous versions did. The one area for which the document does insist on conformance is accessibility.

Changes in the technologies involved are increasingly rapid, which makes it difficult to prescribe preferred versions of browsers or content standards, as was done in previous versions of this document. The improvement in infrastructure available to schools has meant that this document also no longer gives recommendations on optimising the performance or display size of digital resources.

A separate web browser specification is produced every six months by ESA which uses data collected from the Scootle website to monitor web browser usage. The data is used to make decisions on the browsers to be used for testing for the next six months. The revised web browser recommendations will be updated in these guidelines in January and July. The full guidelines will be updated and made available each year in July.

ESA monitors the technologies and standards adopted by the educational jurisdictions and those deployed in classrooms to ensure that these guidelines remain effective and relevant.

1.1.1 ESA testing of content

Formerly, ESA developed a substantial amount of content internally, and dedicate considerable resources to testing that content for usability and accessibility. Previous versions of this document reflected the intensive quality assurance processes surrounding NDLRN content that ESA was responsible for.

With the shift to externally sourced content, ESA is no longer undertaking exhaustive testing of NDLRN content, and these guidelines reflect expected best practices around resources—without ESA necessarily undertaking to police those guidelines. ESA continues to undertake quality assurance, with regards to accessibility in particular, for resources which have been produced for an ESA project.

ESA maintain and can test in the following assistive technologies

Application	Windows 7	OS X
Jaws reader v14	Tested	
ZoomText v9.1	Tested	
MAGic v11.4	Tested	
Dragon Naturally Speaking Pro v12	Tested	
VoiceOver		Available
Microsoft Magnifier	Tested	
Microsoft Narrator	Tested	

ESA tests in the following browsers as needed (list as of July 2014):

Browser	Browser release date	Windows 7	Windows 8	OS X v9 Mavericks	iOS v7.1
Internet Explorer 9[1]	March 2011	Test			
Internet Explorer 10[2]	October 2012	Optional	Optional		
Internet Explorer 11[3]	October 2013	Optional	Test		
Firefox[4]	Monthly	Test	Optional		
Chrome[4]	Monthly	Test	Optional		
Safari 7.0[5]	October 2013			Test	
Safari 7.1[5]	March 2014				Test

Associated documents

These guidelines should be read with the following Education Services Australia documents:

- Educational value standard for digital resources
- Business principles for national sharing of digital resources
- Accessibility specification for content development
- Learning object repository access and exchange (LORAX).

2.0 Principles

Use of the *Guidelines for online content* assists developers to conform with the following four principles for content used across all education jurisdictions:

- **interoperable**
- **flexible**
- **robust**
- **scalable**.

Using the *Guidelines for online content* with the *Accessibility specification* will also ensure that resources are **accessible**.

Resources must also provide **educational value** for Australian students and teachers. This requirement is critical, and is covered by the *Educational value standard for digital resources*.

Interoperable

The aim of this principle is to ensure that online content can be used on a wide range of software and hardware platforms and mobile devices. It assumes that online content will be used within operating environments that support internationally adopted standards and specifications.

Flexibility

This principle relates to effective use of one source of content that can be found and presented in a variety of formats, enabling it to fit into multiple applications, environments or uses. Flexibility is realised by separating content from presentation (by using style sheets for example) and ensuring that online content is described and packaged so that it can be found and recontextualised.

Robust

This principle aims to ensure that web content can operate with a wide variety of user agents (such as assistive technologies) and withstand technology changes without requiring redesign or recoding. The principle is often realised by encoding resources in formats that can be easily interpreted and translated. For example, images may need to be provided in different resolutions for use on different devices or tables simplified for reading in screen readers.

Scalable

This principle relates to using technologies that allows online content to handle increased use and to be capable of expanding to meet growth.

Accessible

The W3C Web Content Accessibility Guidelines (WCAG 2.0) provides a structure to help developers create accessible content. At the top level there are four guiding principles. Content must be:

1. **Perceivable** – Information and user interface components must be presentable to users in ways they can perceive with at least one of their senses.
2. **Operable** – Users must be able to operate the interface and it can’t require interactions that the user cannot perform.
3. **Understandable** – Users must be able to understand the information as well as the operation of the user interface.
4. **Robust** – Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

Using the four principles as the primary focus of the standard makes it independent of any technologies used. Within the principles is a total of 12 Guidelines which define the requirements to be met and the ‘success criteria‘ for each. The success criteria define the level that the requirement conforms to. There are three levels of conformance available – levels A, AA and AAA. The ESA Accessibility Policy requires conformance with levels A and AA.

WCAG 2.0 also introduces the concept of ‘accessibility supported’ content. Accessibility functions can be available through the use of assistive technologies (such as screen readers or magnification applications) within web browsers and must be supported by the users' assistive technologies. This provides some flexibility to users with disabilities but requires that the webpage markup and any functions are consistent in behavior over a range of products.

The standard also allows for content that is not accessible to be available if the content is also available in an alternative accessible format. The standard does require that an 'equivalent' experience is available from the alternative formats so it not usually an easy means of providing the accessibility required.

WCAG 2.0 has been adopted by the Australian Government as the standard to be applied under the *Disability Discrimination Act 1992* and the *Disability Standards for Education 2005.* The Act is enforced by the Australian Human Rights Commission (AHRC).

- For more information on accessibility requirements see the Accessibility specification for content development

3.0 Information model

Online content that contains a number of components grouped into a package for upload and delivery from a repository must conform to the information model described below. The repositories can store websites, multimedia and interactive content. This applies to content provided through LEX and SHEX. MEX and PREX only consume metadata about resources, so only the metadata component of the following is relevant to those pathways for content.

Learning content model

The information model for digital curriculum resource content (both interactive and non-interactive) is illustrated in Figure 1 below. Digital curriculum resources contain resources, organisation information and metadata.

- **Resources** are files and subordinate learning content that are used to facilitate learning experiences. Resources may be any of the file types described in the 'Content requirements' section of the document including, for example: XHTML files, CSS style sheets, MP3 audio files and JPEG images.
- The **organisation** information describes one or more structures or 'organisations' for a package and is provided as an XML file which defines the pathway or pathways through the learning resource. For example, a digital curriculum resource may have two educationally equivalent organisations: one optimised for visual learners and one optimised for use by visually impaired learners.
- Within learning content, **metadata** is structured information about the learning content and its resources. For example, metadata may include the learning content's title, description and educational purpose. The metadata required is described in the ANZ-LOM 1.12 (August 2014) metadata application profile.

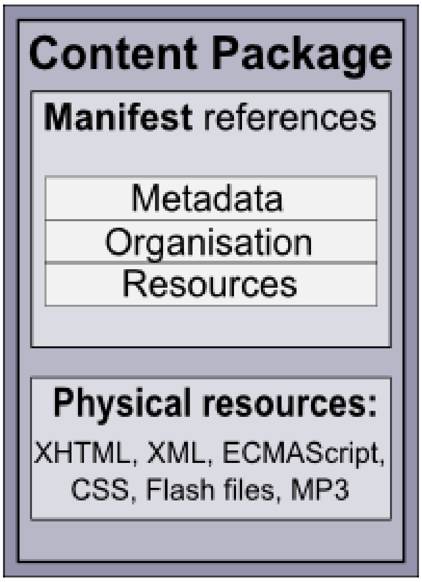


Figure 1: Example digital curriculum resource Learning Content Information Model

Packaging digital curriculum resources

The NDLRN delivers digital curriculum resources to stakeholders as IMS Content Packages (see Content Packaging Specification on the IMS Global Learning Consortium website). The NDLRN has delivered content using SCORM in the past (2004 2nd edition version 1.3 Content Packages, see the Advanced Distributed Learning website); however content providers should use SCORM only by arrangement with ESA.

Resources can be uploaded to digital repositories operated by NDLRN to enable discoverability and distribution to the education jurisdictions. Content is stored on the Sharing Exchange (SHEX).

Content packaged for delivery to digital repositories should have an internal structure that allows intuitive navigation. In particular, it is recommended that the top level of the directory structure in the content package contain either an HTML file named index.html, or an XML file named index.xml, which can act as the common starting point for interacting with the digital resource. It is recommended but not required that subdirectories be used to separate different modules or types of file (e.g. media files).

All file names in the content package should be usable in URLs and portable between operating systems. To ensure this we recommend the following in file and director names:

- Use lowercase alphanumeric ASCII characters (a-z, 0-9), dash “-“, and underscore “_”
- Use slash “/” to separate hierarchical parts of a file path, rather than colon or backslash
- Avoid using the space character

Contributing metadata records

The NDRLN also delivers descriptions of digital curriculum resources, enabling discovery of those resources—but without the NDRLN delivering the resources themselves to stakeholders. This occurs because the resources are hosted externally to NDRLN, either as commercially available, or freely available resources. Resource descriptions are available for these resources as metadata records, from the Publishers Exchange

(PREX) and the Metadata Exchange (MEX), respectively. The metadata descriptions in PREX and MEX are in the same form as those in SHEX.

MEX records can be harvested automatically from content repositories, and this is the default for PREX records. However MEX records are mostly created manually, either by ESA staff, or by content providers using document templates. Please contact the Scootle help desk help@scootle.edu.au for more information on how to contribute your metadata records to MEX or PREX.

Separating structure, presentation and functionality

As much as possible, NDLRN resources must be constructed using technologies that separate the resources' structure, presentation and functionality.

The **structure** of a resource refers to how it is organised. In print media, information in a document may be organised by chapters, and usually includes an introduction and a table of contents. In web media, information in a document may be organised within elements and nodes in an XML file.

The **presentation** of a resource refers to how the resource is rendered. For example, a document may be represented in print, as a web page or as an audio file (for instance as synthesised speech). The presentation of a resource may also be affected by the application of another resource. For example, a CSS or XSLT may alter the presentation of an XML file.

The **functionality** of a resource refers to how the resource interacts with a user. For example, a web page may contain links that open other web pages in separate windows, and a programmed simulation of a calculator may perform calculations in response to user interaction.

Where possible, the resource structure, presentation and functionality should be separated.

Separating web content structure, presentation and functionality enhances interoperability through:

- Flexibility in allowing structural changes in response to changed curriculum frameworks or education priorities without impacting functionality
- Robustness, where the structure persists after presentation elements are impacted by changes to end-user platforms
- Scalability, where added features or services do not impact presentation or structure
- Accessibility, where functionality is preserved as styles are adapted for universal design

It allows graceful transformations of content into different renderings for different devices and easy repurposing of content for different users. Digital curriculum resource structure, presentation and functionality should facilitate the development, maintenance and repurposing of content.

4.0 ESA specific requirements

All websites, digital resources and apps must be developed to meet WCAG 2.0 level AA (double A) requirements.

The preferred format for the presentation of web content is HTML5. Assistive technologies such as screen readers are built to use HTML based content and the content in HTML5 is also available to search functions within the website.

All websites should be tested for accessibility and a compliance statement included which includes the level of compliance and specifies any content that does not meet the requirements. This applies to internal and external client websites.

Discoverability in Scootle

In the following section, we discuss good practices for making your content discoverable on the Web at large, by incorporating metadata within web content. However to make content discoverable on Scootle, and other such education portals, you need to ensure the quality of the separate metadata record associated with the content in NDLRN systems, whether through MEX, PREX, or SHEX. Scootle and other such portals index the metadata records provided to those systems. Not all portals index the text contained in the resources themselves (Scootle for example does not, so metadata within web content will only help your content be discovered on the web, but not on Scootle.)

As noted, the metadata descriptions of resources should follow the ANZ-LOM 1.12 (August 2014) metadata application profile. To ensure discovery of your resources in Scootle in particular, you should add the following information to your resource's metadata. Most of this information is not mandatory in ANZ-LOM, but is required for effective resource discovery in Scootle. (The following description presupposes familiarity with the LOM metadata format. Please consult with ESA staff if you are providing metadata through templates or the MEX user interface, and need clarification.)

- The *general/title* of the resource must contain the expected search terms; search results are ranked higher if a term matches words in the title
- The *general/description* of the resource must be succinct, and contain expected search terms. The general/description of the resource is used to generate the description of the resource that is displayed in search results.
- The resource metadata must include any *targeted year level(s),* encoded following the guidelines in ANZ-LOM: `classification, purpose="educational level", source="Australian School Level", taxon ID = <URL from vocabulary.curriculum.edu.au/ schoolLevel,` ***`without`*** `filename extension (i.e. without ".html" or ".rdf")>`". Year level is used to filter search results, and is required for automated alignment to the Australian Curriculum.

- The resource metadata must include any *targeted learning area(s),* encoded following the guidelines in ANZ-LOM: `classification, purpose=”discipline”, source=”Australian Curriculum Framework”, taxon ID = <URL from vocabulary.curriculum.edu.au/framework, without filename extension>”`. While ANZ-LOM does not require the use of a discipline classification, this is recommended for Scootle. Learning area is used to filter search results, and is required for automated alignment to the Australian Curriculum. (Note that Scootle translates the Australian Curriculum Framework values to a legacy codeset for its internal use.)

- The *education/learning resource type* must be provided, encoded following the guidelines in ANZ-LOM (§5.2). Learning resource type is used to filter search results, and is a prerequisite for automated alignment to the Australian Curriculum.

- The resource must be tagged with relevant *ScOT terms*, under classification, encoded following the guidelines in ANZ-LOM: `classification, purpose=”idea”, source=”Schools Online Thesaurus (ScOT)”, taxon id = <URL of ScOT term, without filename extension>`. The ScOT vocabulary drives the search for resources by topic, and the automated alignment of resources to the Australian Curriculum.

- To ensure discovery of the resource against specific *content descriptions* in the Australian Curriculum, tag the resource with those content descriptions, encoded following the guidelines in ANZ-LOM: `classification, purpose=”educational objective”, source=”Australian Curriculum Content Description”, taxon ID = <URL of content description, without filename extension>`. Resources tagged with specific content descriptions are ranked higher in searches against the Australian Curriculum than resources aligned automatically via ScOT.

5.0 Websites

HTML and formatting

The structure and metadata of a page are critical to the discoverability, use and durability of a website and to the navigation of the site. It sounds basic but it is very important that each page is uniquely identified and has metadata attached to it. The introduction of HTML5 has made this more important because page contents can vary so much.

5.1.1 HTML

HTML5 is the recommended markup language for all web content. A ‘recommendation’ version of HTML5 (version 5.0) is planned for release before the end of 2014. This version will be released by W3C as a ‘complete and fully interoperable implementation’ and will only have minor revisions while version 5.1 is developed for release by the end of 2016.

Web content should conform to HTML version 5.0 requirements.

5.1.2 Navigation

The navigation used should be consistent over the whole site. Always assume that the navigation will provide the primary means for users to find the information they need, so use simple, clear names for navigation items. All navigation items should have a unique name and the text for each must be available to a screen reader.

Many users will not be able to use searches to find content, and will rely on the navigation instead. So it is important that a user has a breadcrumb trail, so that they can always move backwards without having to use the back button of the browser. A site map should also be provided so that users can quickly understand the structure of the site.

5.1.3 Metadata

The HTML metadata standard specifies requirements and recommendations for embedding metadata—i.e. information about a document—in HTML pages, either through the `<meta>` and `<link>` elements, or through commonly accepted microformatting conventions, including microdata (as used by schema.org). The provision of accurate metadata in HTML is essential to the success of any site. Metadata significantly improves search results and helps users find relevant material; absent, inconsistent or inappropriate metadata is a barrier to users.

Pages should include description metadata, to summarise the contents of the page:

- HTML encoding: `<meta name="description" content="…">`

- Microdata encoding: `<div itemscope itemtype="http://schema.org/Thing">…</div>`

5.1.4 Title

The title of any HTML page should be readable by users. It used by both internal and external search engines and is displayed in search results: it is given extra weighting in determining relevant results to a search query, is an important way for users to assess the relevance of a page. Every page should have a unique title, which is concise and descriptive.

5.1.5 Copyright and license

If a page is copyrighted it should link to the copyright statement in the head of the document, using the standard W3C defined copyright 'link' element type. If a page includes a link to the copyright or terms and conditions in the body of the page, the link should contain a `rel="copyright"` attribute.

If a page is released under a particular license, such as Creative Commons, the license statement should be linked through a rel="license" attribute.

Beside these machine readable expressions of copyright/licensing, pages should also display the copyright/licensing status of the page following usual practice, in the page footer where feasible, and with the expected license icon or copyright symbol.

Note that NDLRN content should be made available to schools with as few restrictions as possible. Digital resources should be cleared for use by all Australian jurisdictions and higher education institutions delivering pre-service education. Digital resources should not incur ongoing copyright payments by jurisdictions. Where possible, the strong preference is for content to be licenced for universal public sharing through Creative Commons licence or equivalent.

5.1.6 Other metadata

HTML authors are free to use any other metadata elements which suit their purposes. Where possible it is better to use elements drawn from an existing standard. For NDLRN, HTML authors should give priority to the Dublin Core standard, and LRMI (Learning Resource Metadata Initiative).

Web Page Structure

5.1.7 Organisation

Keep information concise and structured. Avoid using long pages, break content into multiple pages and units and keep the interface as predictable as possible.

Users with disabilities may be using assistive technologies such as screen readers or braille converters to read a page so the pages need to follow a logical structure. Establish a consistent order for your content and maintain that through the site. A screen reader for example will start at the top of a page and read to the bottom so you should have the content available in that order, rather than controlling the order of rendering using the DOM or CSS.

5.1.8 Headings

Headings should follow a logical sequence starting with h1 to a maximum of h5. Use the headings to provide structure to the document. Don't use them to provide decorative or graphic components. Users of screen readers will scan through the headings to locate the content they are interested in, so the text in the headings should be descriptive so that a user can quickly understand the content.

If you are using graphics for the headings (not a recommended option), you will need to ensure that the text of the heading is also available to a screen or braille reader.

5.1.9 Menus

If drop-down menus are provided they should be built with tolerance for users who have problems with operating a mouse. Once selected a drop down menu should stay open for a few seconds. If tolerant drop-downs cannot be used it may be better to just use one level of drop-downs.

Users must be able to use the keyboard to tab to drop-downs and then tab through them (rather than having to use the down arrow key).

5.1.10 Style sheets

The main benefit of using CSS is to separate content from its presentation. *Content* should be defined in the HTML and *presentation* should be defined using CSS. Structuring the site in this way enables users to disable or override the CSS while leaving the content fully accessible.

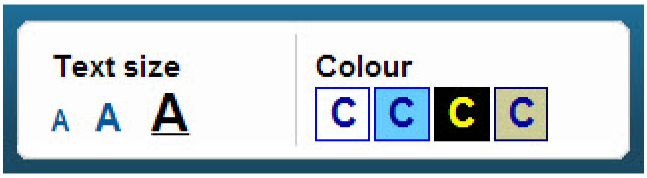
The order of presentation of the content of the pages should match or be very similar to the visual presentation (i.e. reading from top to bottom and left to right) rather than using CSS to order the content. This will ensure that a screen reader will read the content in the order it is presented (screen readers ignore the CSS).

Do not use CSS to display content such as background images that include content or control presentation in a way that affects meaning.

Websites should include user selectable CSS to enable users to enlarge text and choose different standard colour and contrast options. Standard format for these CSS is to provide three text zoom levels (100, 150 and 200 percent) and three page styles in addition to the default style –

- dark blue text on a light blue background
- black background with light blue text and yellow navigation highlights
- yellow background with dark blue text.

A selection panel should also be provided to enable the users to switch the styles. An example panel is shown below.



5.1.11 Mobile devices and tablets

All websites must be developed for use on desktop computers, tablets, and mobile devices. The websites must meet the WCAG 2.0 level AA requirements (except those not relevant for mobile devices, such as those that specifically address assistive technology). We recommend that websites also refer to the W3C Mobile Web Best Practices for recommendations on how to effectively deliver content to mobile devices. Many of the requirements around consistency of navigation, design, formatting etc apply across all of the viewing mediums, but there are a number of requirements specific to mobile devices as listed below.

We recommend that specific style sheets be used to manage the display of content on tablet devices. Because of the display restrictions on mobile phones and the variations in size, it is usually easier to develop a mobile version of the website rather than rely on style sheets. The decisions on how to manage the mobile devices website will be dependent on where your users will access the site.

ESA are currently only testing using Apple devices as they represent the overwhelming majority of our users of mobile devices. Other devices may be added to testing as their market share increases; but content providers should as a matter of course test against Android as well as Apple.

All websites must be optimised for viewing on mobile devices and use CSS specifically for tablets and ensure that all content will work on mobile devices. Care must also be taken to ensure that images and navigation items can display correctly on mobiles, and that navigation using selection or drag and drop functions are available on the mobile devices.

5.1.11.1 Editorial

Include a link to the desktop version from the tablet and mobile versions of the site
Provide a link to the desktop site on the tablet and mobile versions, and to the tablet and mobile versions on the desktop site.

Users should not be forced to use a tablet or mobile version on their device, and should have the option to switch between the desktop, tablet or mobile version. Some users with disabilities like to use the tablet or mobile version of a website on a desktop computer (because of the larger sizing of components or the simplified navigation), and should not be restricted from using them.

5.1.11.2 Design

Provide sufficient read and select space for touch screens
Content must be large enough to read, and must have a large enough selection area. Provide padding around links so that they are easily selected.

Touchscreens targets should be at least 9.6mm across
All selectable icons and controls must be at least 9.6 mm across (equal to 44 pixels for standard resolution and 88 pixels for high resolution). This is a minimum size so where possible use larger icons and controls or increase the padding around them to ensure a large touch area.

An inactive space of at least 1 pixel must be provided around each target
Ensure that margins or some other mechanism for spacing exists around each selectable icon or control item. Sometimes text big enough to read is too small to touch. Large touch zones are especially important to enable better accessibility to links for users who may have dexterity problems or who cannot see the screen, and who may not touch with as much accuracy as a sighted user.

5.1.11.3 Navigation

Swipe areas must be clearly indicated
Content is often misunderstood or not discovered as users don’t realise they can swipe to reveal content.

Indicate that carousel content can be changed via swiping by adding arrows on the left and right edge. Also add alternative text to these arrow images, instructing users that the content can be changed by activating the arrow image button elements.

5.1.11.4 Development

Content must degrade gracefully when the styles are unsupported or removed
Content must be readable and the website must function without the CSS working. When background colours, layout, or features are missing, the user must still be to able read content and access the functionality. Users of older mobile devices may have poor support for fonts, colours and styles, so consider this when setting default values.

All actionable page elements should have a visible focus

When focused, all focusable elements must have a visible outline so that users can track where they are in the content. Ensure all links and focusable elements do not have their outline suppressed in the CSS.

Actionable page elements must be distinguishable from non-actionable elements
Users must be able to determine visually if an element is actionable or is simply informational or static content. All links for example should have underlining or other visual attributes to indicate that they are selectable.

5.1.12 WAI-ARIA

WAI-ARIA, the Accessible Rich Internet Applications Suite from the Web Accessibility Initiative, is a W3C recommendation published in March 2014, which makes dynamic content and advanced user interface controls more accessible. It is realised by adding new attributes to HTML, which describe properties of elements useful for accessible design. ARIA can provide functionality such as navigation short-cuts or drag-and-drop to users with disabilities. Many web creation systems have ARIA frameworks built in so it can simply be a matter of configuring the functionality to suit.

ARIA also allows websites to update without reloading the page, which can be an issue for a screen reader. Ensure that any important information that has been passed back to the page is available to a user by using the alerts available, and use the timer functions to alert users to timing constraints applied to actions such as completing forms.

ARIA can also be used to assist in passing menu information to screen readers and to provide ‘landmarks’ within pages to assist with navigation. As more ARIA functionality is added to libraries, it is important to test that the functions used are compatible with our recommended web browsers.

5.1.13 Tables

Tables should only be used for presenting tabular content.

Screen readers treat tables differently, and will read the table in a consistent order if it is formatted correctly. All columns and rows should include table row (`<tr>`), table head (`<th>`) and `scope` tags to assist the screen readers.

The option to skip a table (i.e. linking at the start of the table to the text after the end of the table) should also be provided: users using screen readers will get limited use from tables, and do not have the option of straightforwardly “scrolling past them”.

5.1.14 Skipping navigation

Unless a screen reader is told to skip the navigation at the top of the page, it will read it on each page. To get the screen reader to skip the navigation, add some code to each page above the navigation, as below:

```
<span class="hidden">
       <a href = "#content">Skip to content</a>
</span>
```

Before the content add:

```
<a name="content"></a>
```

If the page has local navigation you can also get the reader to skip to that by adding the following before the page navigation:

```
<span class="hidden">
       <a href = "#pagenav">Skip to page navigation</a>
</span>
```

Before the local page navigation add:

```
<a name="pagenav"></a>
```

5.1.15 JavaScript and Scripting

Under WCAG 2.0 sites can operate using JavaScript, but the scripting must be accessible. It is important to consider that some users will disable JavaScript or may be using browsers that don't fully support scripting. If your web page or application will not work without scripting, you must avoid confusing or non-functional content that appears to function, but does not because of lack of JavaScript support. If this cannot be achieved you will need to provide equivalent functionality or material on an alternative page that is accessible.

All JavaScript based navigation should include alternate navigational images or text links within `<noscript>` tags.

The `href` attribute should be used for a direct link to content, either in the parent (`target="_parent"`) window or a new (`target="_blank"`) window, so non JavaScript enabled browsers can access the content.

Use keyboard event handlers in addition to mouse event handlers, since not all users will be using a mouse: e.g. `<onclick=openWin(index.html) onkeypress=openWin(index.html)>`

When using JavaScript event handlers, return false so that the browser does not attempt to open the link in the associated <a href> element: this is confusing to users on assistive technology.

5.1.16 Pop-ups

The use of pop-ups is not recommended because they present too many issues for their management by users with keyboard only access.

If they need to be used, it is important to consider how a user will gain access to the pop-up, and how they will be able to close it. The user must also be provided information on how to manage the pop-up (e.g. keystroke to be used to close the pop-up).

When pop-up windows are invoked to display separate html content, use the `‘onclick’` and `‘onkeypress’` event handlers to open the window (both returning false so the `href` is not followed by JavaScript enabled browsers), and point the `href` attribute directly to the html content, targeting a `"_blank"` window. This will allow JavaScript enabled keyboard users to benefit from JavaScript window control, while allowing non JavaScript enabled users to view the content, albeit in a full browser window.

```
<a href="index.html" target="_blank" onclick="openWin(index.html); return false;"
onkeypress="openWin(index.html); return false;">
```

5.1.17 Frames and iframes

The use of frames and iframes to structure a website is not recommended. Not all screen readers will read frames and iframes, and those that do begin at the first frame they find and read through it completely before moving to the next frame and reading through that.

If frames are used they must be declared in the page header, and named informatively, with names such as ‘main banner’ or ‘content’. Users using screen readers can then scroll through the list of frames and find the frame they are interested in reading. The same rules apply to iframes.

Finding information in frames and iframes can be very frustrating for users of screen readers. If a whole website is developed in frames you should identify and suggest to users the screen readers that you know can read the site. If only individual pages are created using frames, it is best to provide a version of the page that does not use frames and then link to it.

5.1.18 Hiding text

If you have a need to hide text so that it will not be read on the web page but will still be available to a screen reader (very handy for long descriptions text for example), it is best to position it off the screen using the CSS. That is because other CSS conventions for hiding text will also suppress the text from a screen reader. An example of how it can be done in the CSS is:

```
.hidden {
       Text-indent: -5000px
}
```

In the HTML you would then add:

```
<p class=”hidden”>Text will be positioned off screen</p>
```

5.1.19 Access keys

Most web browsers now support access keys so that users can move between pages or to elements of a page by hitting certain key combinations. Implementation within browsers varies considerably, and if access keys are used a user would need to spend some time becoming familiar with the combinations used within their browser.

Implementation also varies on what activating the appropriate access key keyboard combination will do. Some simply set focus to the element that has the access key (the user must then press Enter to activate it), whereas others will immediately activate it. This varies even more depending on the type of element – links, buttons, and form controls may behave differently.

One of the biggest concerns with access key shortcuts is the possibility of the access key overriding the keyboard shortcuts of screen readers, which have many more keyboard commands than standard browsers. This is of primary concern because screen reader users are usually reliant on keyboard accessibility and are thus a primary beneficiary of access key functionality.

In cases of conflicts, the screen reader shortcuts generally take precedence, meaning that the access key shortcuts are effectively disabled. The accessibility benefit of access key shortcuts is lost, but screen reader functionality is left intact.

Our recommendation is that until more standards are applied to the use of access keys, they should not be implemented by websites.

5.1.20 Timed tasks and instructions

Some users with reading difficulties or disorders such as Attention Deficit Disorder (ADD) will require additional time to complete tasks, or will need to take breaks to regain concentration when doing tasks such as completing forms, viewing slide shows or progressing through multimedia content. If tasks have time limits, provide users with the option to extend the time allowed, and give them plenty of warning if a process is about to time out so that they can extend it.

Consider breaking forms into small sections and allow users to save the form as they progress, or auto save the information. Provide a progress bar so that users know where they are in completing the task, and ensure that any validation of the data entered is done as the form progresses, and that any input issues are clearly indicated and display where the data is required.

If you have instructions or information that must be read before a user can progress, ensure that allowance is made for users with difficulty reading and fully comprehending them. Keep any instructions as concise as possible, and where possible use lists rather than a block of text. Consider using video or audio instructions if providing written instructions becomes too long. Ensure that a text description is provided with any video or audio instructions, and that the text description provides clear instructions that extend beyond a straight transcript of the content.

5.1.21 CAPTCHA

A CAPTCHA (or Completely Automated Public Turing test to tell Computers and Humans Apart) is used on forms to deter bots from spamming the website. Users are asked to enter a series of letters or words from an image. A CAPTCHA is a problem for a screen reader user because the images used are created dynamically and the letters cannot be included in the alt text.

A number of fall back options can be provided. The first is to provide an audio CAPTCHA. A screen reader user would play the audio of the voice reading out the numbers and letters and then enter them into the form. However audio CAPTCHAs can be as hard to make out for users as visual CAPTCHAs are (audio CAPTCHAs add background noise for the same reason visual CAPTCHAs distort letters—to prevent automated recognition). Another option is to provide an alternative to the CAPTCHA, such as requiring the user to submit an email address or some structured data (such as a series of numbers) into a form.

Providing a challenge response such as a question that a human would find easy to answer but a robot could not is another option. The questions provided could have obvious word answers or a combination of numbers and the questions should be rotated regularly. Avoid questions that blind users may have difficulty in answering, such as questions relating to colours or shapes that need visual interpretation.

5.1.22 Flash

The use of Flash should be avoided, not just because it is a problem in terms of accessibility but also because it does not function on Apple mobile devices. Consider other options in preference to using Flash for slide presentations, videos or interactive items (such as HTML5, JavaScript and CSS), and avoid using Flash to provide site navigation.

If you do use Flash you will need to provide a fall back for Apple mobile devices; use the accessibility tools available in the authoring applications, and always provide low bandwidth versions. It is not possible to make some Flash content such as games accessible to blind users; but if you consider the design, structure and value of many interactive items, it is possible to make them accessible using products other than Flash.

Design

The role of the designer is very important in creating an accessible website. Design considerations such as colour contrasts, allowances for the zooming of content, the relationships between text and images, and the use of consistent navigation structures and components all assist in making the website accessible.

5.1.23 Design components

The key aspect of designing an accessible website is accommodating a wide a range of users. Considering the needs of the intended users and the types of information that will be delivered is critical. Aim to provide users with an easy to use design and a consistent navigation.

The design should use a heading structure that has styled headings ranging from H1 to H5 that conform to contrast requirements. All fonts must use relative sizing units (such as percents or ems) rather than absolute units (such as pixels or points). If background images are used with the headings, ensure that they are used as background images so that they will zoom correctly with the heading text and not interfere with the heading text in screen readers.

Do not use images of text. All text should be included in the HTML and styled to display as required rather than presented as an image.

Consider that users of zoom functions in the browser or of zoom tools will be viewing only parts of the screen at any time. It often assists them to understand where they are on a page if the visual design allows for clear distinctions between the components of a page.

5.1.24 Colour and contrasts

Care needs to be taken with how colour is used and the contrast between colours. This is most obvious with the use of colours in type but it also needs to be considered within graphics, diagrams, maps and charts. It is recommended that colour schemes be tested using a colour analyser before the design is finalised to ensure that the colours selected have good contrast with each of the combinations in which they are used.

Colours cannot be perceived by users who have low vision or are colour blind. Users with lower end handsets with poor support for colours may also have trouble distinguishing colours.

5.1.24.1 Provide sufficient contrast

WCAG 2.0 level AA requires a contrast ratio of 4.5:1 for normal text and 3:1 for large text. Level AAA requires a contrast ratio of 7:1 for normal text and 4.5:1 for large text. These contrast ratios are specified to assist people with visual impairments, colour deficiencies, learning disabilities, and cognitive impairments to read and comprehend the content.

These contrast rations also apply to the use of colour as a differentiator, for example when colour alone is used to indicate the presence of a link or selected tab with alternative text. The colour difference between the link text and non-link text must also meet these requirements. If a website uses background colours then also set a foreground colour, to ensure that if the background colours are turned off that the text is still readable.

All images that use colours to convey information should also be checked for colour contrast. Maps, diagrams, graphs or images with text should all be checked. A key should be included on all images that use colour to provide information.

5.1.24.2 Provide alternatives for colour conveying information

Colour used to convey error messages, selection, emphasis, links, and other meaningful information must be used in combination with an alternative such as different line weights, underlines, patterns or fills.

5.1.25 Fonts

Designers are no longer restricted to using system fonts to display text, as web font technology has made a wide range of fonts available. Display of the fonts is managed by the web browser; screen readers ignore the font information and read the text available in the HTML.

Readability should be the primary consideration in choosing a font. Try to keep the style of font displayed as consistent as possible (e.g. don’t mix a bold and all caps font on one heading level with a fine san serif all lower case font on another).

Keep the number of fonts used to a minimum. Take care not to overuse font variations (such as bold, italic, caps and small caps, all caps etc) for emphasis. Keep it simple. Many users with disabilities have difficulty if they need to interpret different fonts. Don’t use blinking or moving text as they can be difficult to read and create a distraction for many users.

5.1.26 Headings and text

All headings should be included in the heading structure used for the site. This structure should include a maximum of six levels of heading. If background images are used with the headings then the images and the heading should be included in a <div>, and the colours specified so that the browser can manage the display when the heading is enlarged when zoomed.

All text should be provided by the HTML for the page. No text should be supplied as images.

The length of sentences should be kept in the 10 to 15 words range and paragraphs kept to a maximum of about five sentences. Most web users find that left aligned text is easier to read, due to the even spacing between words. Justified text can cause issues for users with dyslexia because it can create ‘rivers’ of white space (created by the computer when justifying the text) that can be very distracting to them. Ensure that a suitable space is allowed between lines (leading), so that users can easily distinguish between lines when reading. Lines should be spaced at least 1.5 times the font size. Line length in the default font size should be between 50 and 70 characters, and text must be able to resize up to 200 per cent without requiring the user to scroll horizontally to read a line of text on a full-screen window.

5.1.27 Zoom

Most web browsers allow a web page to zoom to at least 200 per cent. Some enable text only zooming, but most enlarge all content in the browser window, and re-stream the text and reposition the enlarged images. Assistive technologies such as ZoomText allow enlargements to 600 per cent, but enlarge the existing screen rather than enlarging the actual page components.

It is important to consider the implications of zooming a page, especially with items such as tables and forms. Relationships between components should be maintained so that when a page is zoomed the relationships between labels and input fields is still clear.

Editorial

5.1.28 Alternative text

Alternative text (or alt text) should be supplied for all non-text items including images, charts and diagrams. Alt text should be kept succinct and describe the content and function of the image. The text should not repeat any captions, and it should not repeat text that might be included in the image (or link). It should be limited to about 250 characters.

The alt text is read by screen readers to provide a description of the content and function of an image, and it is displayed in place of the image in browsers if the image file is not loaded or when the user has chosen not to view images.

Images used for decoration only should have empty double quotation marks (“”) added to the alt text field. A text reader will ignore these entries. Any repeated images (such as multiple copies of a logo) can have alt text added to the first instance but empty tags provided for the rest.

5.1.29 Long descriptions

All complex images such as maps, charts and diagrams need long descriptions which describe their content and function. The long descriptions can be as long as required to succinctly describe the content, so that a user who cannot view the item can get the same information as a sighted user.

The long description is created either as a separate web page, which is linked from a ‘long description’ text link under the diagram, or as hidden text in the HTML. The additional page option is preferred as it provides users of screen readers with more options.

6.0 Documents

Formats available

All downloadable documents must be made available in accessible formats and in more than one format i.e. HTML, text, ODF, RTF, Word. Our preferred option is that all content is available as HTML pages. If content needs to be supplied in a document format then that format must allow it to be read in a screen reader or braille display. The minimum requirements are that all pages must be formatted so that all headings are styled, tables are structured so that table headings and content can be identified, all images have alternative text (and long descriptions where required), and the content reads in the correct order.

It is preferable to prepare documents in a format that can be read by a range of applications such as text, ODF or RTF. Consider that all users will not have the latest versions of an application such as Microsoft Word, so prepare documents in a format that can be opened in earlier versions.

PDF documents

The use of PDF documents is not recommended. Our preferred format is HTML so that the content is available to text readers and search engines.

While it has been claimed that PDF files are easy to make accessible, the reality is that making a PDF accessible requires a document structure along with heading styles, alternative text on all images, and all table content to be marked-up. This work must be done when creating the source document. Once a PDF is created it still needs checking and updating in Acrobat Professional before it can be made accessible. The work required to ensure that a PDF is accessible can be very demanding of time and resources.

If a PDF must be used then the information should also be supplied in an accessible form (such as HTML), so that the PDF document is not required to be accessible.

7.0 Images

Alt text

See 5.1.28

Long descriptions

See 5.1.29

Graphs and diagrams

All graphs and diagrams should include alt and long description text.

Keys should be provided that will be clear to all users, including those that are colour blind. Use different line styles, weights or patterns rather than just colour to indicate variations.

Image maps

You must provide redundant text links for each active region of an image map.

When using client-side image maps, the main image must have alt text. The alt text should describe the basic information provided by the image. If the main image does not convey content, but is primarily just a container for the hot spots (e.g., a navigation bar), then an empty alt tag (“”) can be used.

Each image map hotspot (`<area>` element) must have an equivalent alt attribute. Because the hotspots are clickable, each hotspot must have alternative text that describes the function of that hotspot.

Hotspots for server-side image maps cannot be given alternative text and are not keyboard accessible, so they should not be used.

8.0 Video and audio

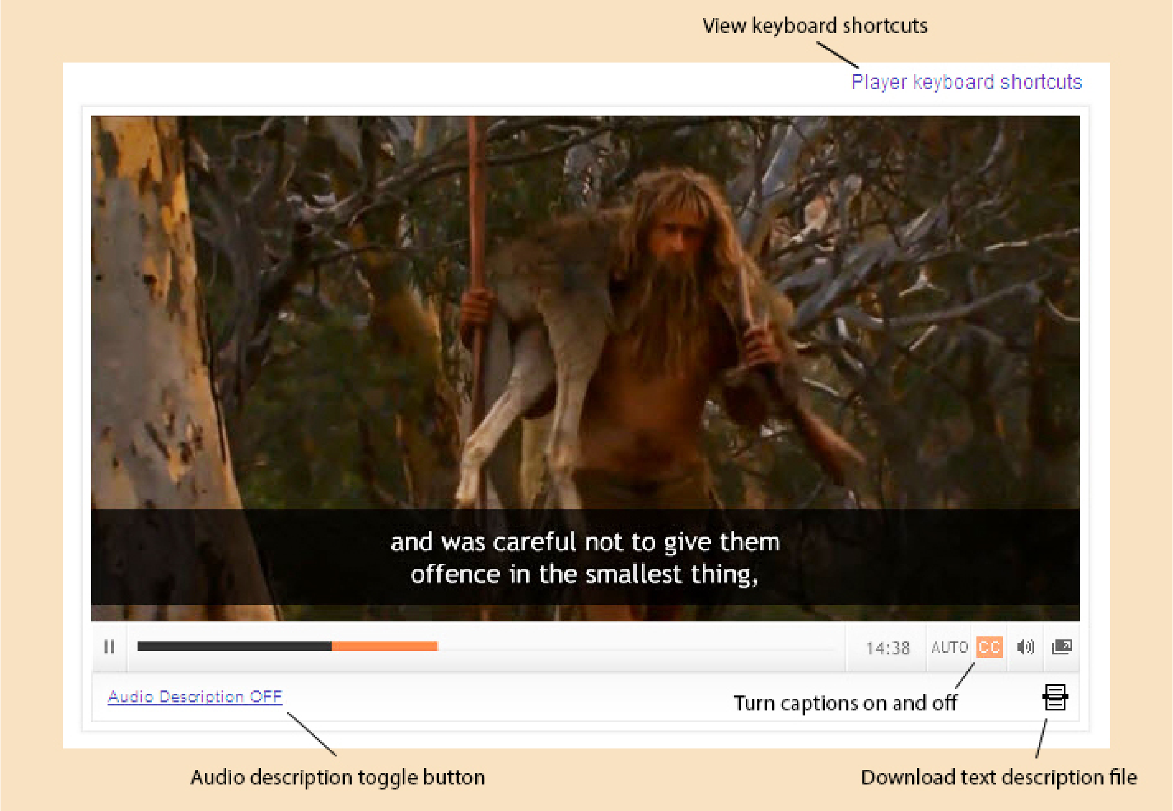
Level AA Requirements

All pre-recorded video content must include captions (Guideline 1.2.2, level A) and an audio description version of the video (Guideline 1.2.5, level AA) to meet the success criteria.

That is the strict interpretation of the standard; but we allow as conformant any video that does not have gaps in the dialogue to enable the inclusion of the descriptions (such as a “talking head” video where the presenter speaks on a subject), so long as a text description is prepared that includes a transcript of the dialogue along with information to put the presenter into context. The text description should be in a .txt format and available for download.

Display of accessible videos

ESA has an agreement with Viocorp for the storage and delivery of accessible video content using their Viostream product and video player. Their system enables the upload of video files, caption files, audio and text descriptions and integration within their player. Controls are provided on the player to control the accessibility components.



View keyboard shortcuts

Turn captions on and off

Audio description toggle button

Download text description file

and was careful not to give them offence in the smallest thing,

Captions and descriptions

Captions are provided for use by those that cannot hear the sound track due to hearing difficulties, for users in environments where listening to the sound may be difficult (in a noisy environment for example), or where the audio may be difficult to understand (such as the use of heavy accents or very fast speaking). Captions provide a text display of the dialogue and other important sound information, displayed synchronised with the video. They can be supplied as open captions (embedded into the video and not able to be controlled by the user), or as closed captions (user can turn them on or off). Captions can be displayed over the area of the video or below the area of the video; the WCAG standard does not specify any formatting or display requirements.

Descriptions provide an audio or text description of the video which can be used by a sight restricted user to understand the content of the video. An audio description uses the existing audio track and inserts descriptions of the video content into gaps in the dialogue. The audio description is then played with the video file as an alternative audio track, when the audio description option is selected. To meet level AA requirements the audio description must synchronise with the existing video track (you cannot insert pauses into the video to add more description content into the audio description for example).

In contrast to an audio description, a text description is provided when it is not possible to fit descriptions of the video in gaps in the dialogue (e.g., for ‘talking heads’ where a dialogue fills the full length of the video). The text description is viewed as a web page or read in a text reader, and is therefore not read or viewed with the video. The text description should provide all dialogue and descriptions of actions and actors in the video. A simple text description for a talking head video might start with the name, role of the presenter and context of the presentation, followed by a transcript of the dialogue. A video with many actors and rapidly changing scenes will be like a script for the video, and include descriptions of the scenes, names and descriptions of the actors involved, and emotions and emphasis in exchanges as well as the dialogue.

The W3C specification makes references to the use of SMIL to deliver accessible video content. This is not recommended as the SMIL functionality is not available on many video servers or players.

8.1.1 Captions

Captions must include all dialogue, describe important sound effects, and where possible provide information on who is speaking. Our recommendation is to use a video player that allows the use of closed captions. The captions are best prepared by an outside supplier and the files supplied in .srt (SubRip Text) or TTML format for loading to the Viostream server.

The .srt file numbers each of the captions and provides timing information and the text for the captions in the line structure in which they will be displayed. A sample .srt is shown below.

```
1
00:00:00,360 --> 00:00:04,220
(MEN SHOUT, HORSES NEIGH)

2
00:00:04,240 --> 00:00:08,180
MAN: For 2,000 years
they stood on guard under the earth,
3
00:00:08,200 --> 00:00:11,820
poised to defend a mighty leader
who dreamed of immortality,
```

Captions supplied as .srt files appear to be just text files, but there is hidden formatting information in the files, so they should only be edited in caption creation and editing applications such as Subtitle Edit or MovieCaptioner.

The same information in TTML is shown below.

```
<body region="subtitleArea">
   <div>
     <p xml:id="subtitle1" begin="0.36s" end="4.22s">
       (MEN SHOUT, HORSES NEIGH)
     <p>
     <p xml:id="subtitle1" begin="4.24s" end="8.18s">
       MAN: For 2,000 years<br/>
       they stood on guard under the earth,
     <p>
     <p xml:id="subtitle1" begin="8.22s" end="11.82s">
       poised to defend a mighty leader
       who dreamed of immortality,
     <p>
   </div>
</body>
```

The Viostream player controls the display of the captions. They are displayed in a maximum of two lines at the bottom of the video area. The format of the display is not able to be changed.

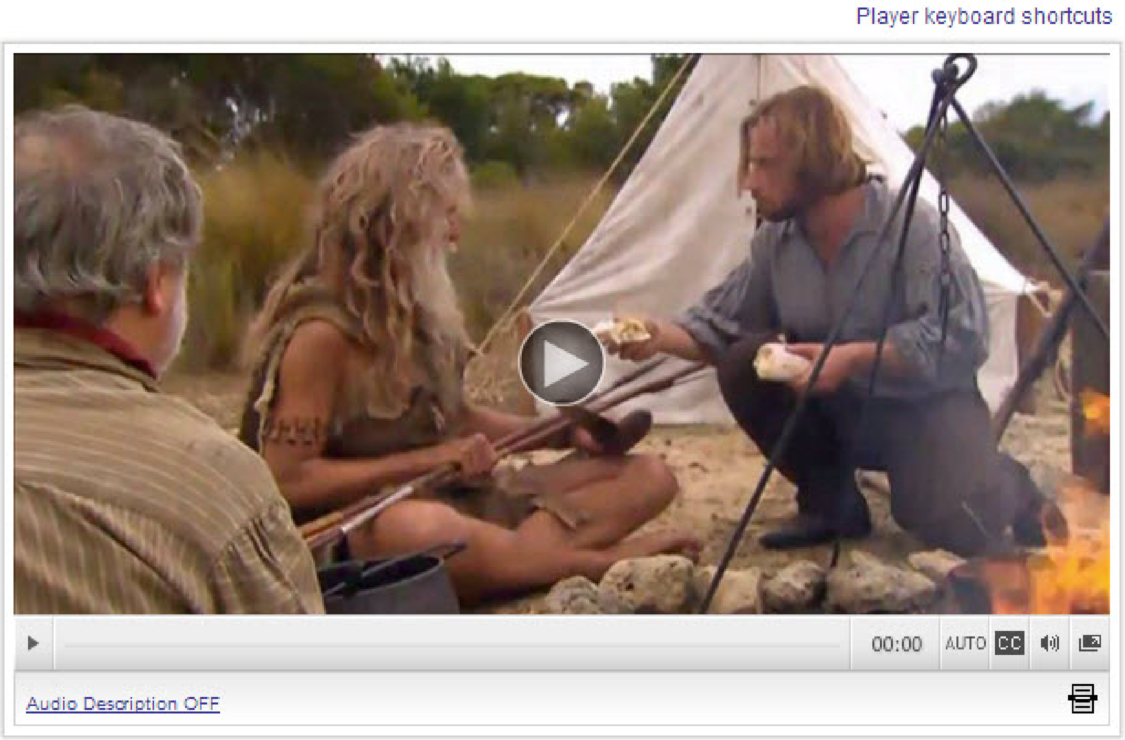
8.1.2 Descriptions

8.1.2.1 Audio descriptions

Audio descriptions (AD's in short) are prepared as a separate audio track which combines the original audio track with descriptions of the video content, written to fit within gaps in dialogue. An audio description will only meet the success criteria if enough information can be included in the descriptions for a user to gain a good understanding of the content of the video.

At AA level the video and audio tracks must remain synchronous. You can add description content over non-essential dialogue that has been muted back in order to fit the descriptions. This does make the preparation of the audio description more expensive but it might be a better option than writing a detailed text description.

Scripts for audio description files are written at the same time as the captions. They include timing information and the text to be spoken. Once the script is checked it is returned to the caption house for recording and combining with the dialogue from the original audio track, and a separate version of the video file with the audio description track as the sound track is created. The AD version of the video is loaded to the Viostream server and then linked to the master version of the video. Once it is linked an “Audio Description OFF” link is displayed on the player. Selecting the link will load the AD version of the video.



8.1.2.2 **Text descriptions**

Text descriptions are prepared as text files and loaded to the Viostream server for delivery as a HTML page. A simple text description can be a transcript of the dialogue with basic information on the speakers of the content of the dialogue. A complex text description could be similar to the script of the video with descriptions of the actions of all actors involved, interactions between actors, sounds, dialogue, and a detailed description of the location in which the video takes place.

8.1.2.3 **Which one do I use?**

The preferred option is to supply an audio description, and is required to meet the level AA success criteria. In videos where the dialogue fills the majority of the sound track it is recommended that a text description be prepared; ESA will then consider the video to conform to the WCAG 2.0 level AA success criteria. Text descriptions provide a better experience for users if the video contains technical information, musical or dance sequences that are better described in a longer form.

8.1.2.4 **Silent movies and music**

- Silent movies can use an audio description, but they must begin with a statement that the movie does not include a sound track.

- Where the whole video is music content, the captions will describe the music in simple terms and provide any lyrics. It is best to provide a text description, as longer descriptions of the music and the lyrics can be included independent of the video timing.

Keyboard access

The video player used must be able to be used from a keyboard. The Viostream player meets keyboard accessibility requirements.

Video players

The video player used must provide the following functions in order to meet the accessibility requirements –

- Navigation using the keyboard
- Control captions
- Display audio or text description
- Enable video to be paused
- Provide forward and reverse
- Allow volume control

Videos must not auto play when loaded. The user must select when to play the video.

Audio files

Audio only files must be accompanied by a text description file which includes a transcript of the dialogue plus descriptions of any significant sounds or actions. The Viostream player can play audio files and text descriptions, and .txt files are used as the format for files to be uploaded.

If you are not using the Viostream player, the player used must be keyboard accessible, have controls to enable users to pause, forward and reverse and change the volume. Audio files must not auto play when loaded. The user must select when to play the file.

Suppliers

ESA will advise content providers on request of captioning services that can be used to meet NDLRN requirements.

9.0 Forms

Forms are used for many types of interactions on the web and present a number of challenges for users with disabilities, particularly those using screen readers, keyboards, zoom tools or assistive technologies. Taking the time to create accessible, well-organized forms helps all users.

Structure

Forms should be structured in a way that is clear, logical and intuitive. Ensure that the tab order is logical and that a user can easily tell where they are in a form.

Provide clear instructions about what information is needed and what users can expect when using the form (e.g. number of pages, any time restrictions). Instructions, cues, required form fields, input requirements, etc. should be clearly identified. If any set formats are required (such as phone numbers), these need to be clearly indicated and a sample provided.

Keyboard navigation

Ensure that forms can be completed using only the keyboard. JavaScript can cause problems with keyboard navigation. If JavaScript is used to manipulate form data, set focus, change form elements or submit forms then the forms should be tested to ensure that the keyboard navigation meets the requirements in a range of browsers, operating systems and on mobile devices.

Forms should not be completed relying only on gesture. Any required gestures must be implemented so as to support accessible alternatives.

Label and field associations

Use clear and explicit form labels so that screen reader users can easily understand what the form element is. The use of explicit form labels is important for all users but essential for users of screen readers to understand the form and understand where they are in the form. Text labels should describe the function of each form control. Use 'title' attributes when visual labels are not possible, and do not solely rely on ARIA labels or HTML5 attributes like 'placeholder'.

Labels assist users understand what input is required, and need to be both visible and audible. Empty spaces between labels and controls can leave users disorientated within pages. This has a larger impact on users with low vision, who scale pages up or use screen magnification on mobile devices.

The `<label>` element is used to associate a text label to a form control. Place the label as close as possible and adjacent to its respective form control (i.e., text box, check box, radio button, menu, etc.). This allows a screen reader to read the associated label text when the user navigates to the form control and to enable users of zoom tools to understand the relationship without having to zoom out. Labels can also be associated programmatically with form controls using HTML markup.

In portrait mode labels work best above the field. However, labels for radio buttons and checkboxes work better to the right of the control. In landscape mode labels for form fields work best to the left of the form field. Labels can also be placed within the text field or drop down menu itself.

Use CSS media queries to remove empty space and position content into mobile friendly layouts. Also ensure that buttons in forms are not distanced from form elements.

Use any field notation inside the explicit label for a form field. If field constraint information (such as a 'required' field notation) is not part of the text label or associated with a field through a method supported by assistive technology, users such as those who use a speech reader may not have access to this required information.

A default input mode must be indicated

Indicating a default input mode for input helps prevent errors when inputting content and ensures formats are correct. Specify text, numbers, language and/or input format if the device is known to support it. There are various means of doing so, including HTML5 type attribute of `<input>`, the WAP CSS `-wap-input-format` attribute, and using scripts.

Form focus

Most screen readers will automatically switch to ‘forms’ mode when focus is shifted to a form element, and back to ‘virtual cursor’ mode when focus shifts to a non-form element. The keyboard can also be used to manually change modes. When a field receives focus it should not initiate a focus change: this may prevent users of the keyboard and assistive technology from accessing the control. Focus should only change when the user presses tab, enter or space on a link or button or clicks on a field, link or button.

Drop down menus in forms should always be provided with a Submit or Go button.

When a field has been completed (such as when a set number of characters have been entered), the focus should not shift out of the field to the next field. If the focus is shifted it will prevent users of the keyboard or assistive technology from verifying information or correcting mistakes. The focus should only shift when the user presses tab, enter or space on a link or button or clicks on a field, link or button.

It is also helpful if styles are used to show that a field is in focus. This can be done by adding a highlight colour around the outside of the focus field for example.

Form control grouping

Controls, elements, and objects must be properly grouped for users of assistive technology such as speech readers to understand the relationship among controls. Additionally, controls must be grouped properly for standard behaviours such as keyboard interaction to work correctly, e.g. radio buttons selection.

Ensure all radio buttons in a group have the same ‘name’ attribute. Other groupings of form controls will require a higher level description (such as "Shipping Method" for a group of shipping option radio buttons). This descriptive text can be associated to the group of form controls using <fieldset> and<legend>. The <fieldset> identifies the entire grouping and <legend> identifies the grouping's descriptive text. Using <fieldset> and <legend> ensures that the text description is read to screen reader users when the grouping is navigated to.

Submit buttons

Provide a ‘submit’ button on all forms. You may use an image to perform this function but ensure that the text in the graphic matches the text in the value attribute and that alt text is provided for the image.

Form image buttons must have an alt attribute that describes the function of the button. Image buttons are often used to provide a more visually appealing or a smaller version of the standard form buttons. The alternative text should describe what the button will do when selected, such as ‘Search’, ‘Submit’, ‘Register’, ‘Place your order’, etc. For instance, `<input type="image" alt="Submit Search">` might be appropriate for an image button on a site search form.

Help and feedback

If an error occurs on a form, the error should be highlighted and an error message displayed. Users of assistive technologies may not be aware an error has occurred until the page is refreshed. It is best to display error messages at the top of the form and indicate the field in error. Move focus to the error message after page load. Ensure that enough information is provided in the error message to enable a user to understand the problem, and to locate the field that needs correcting.

When an input error is automatically detected and suggestions for correction are known, then the suggestions should be provided unless it would jeopardise the security or purpose of the content.

10.0 Web tools

Keyboard navigation

Keyboard accessibility is one of the most important aspects of disability access. Blind users and many with physical disabilities cannot use a mouse or touch pad, and need to use the keyboard to navigate the web. Some will use tools such as head mounted wands or 'puff and sip' devices activated by airflow from the mouth to enable them to use the keyboard. If a web site is not keyboard-accessible, many people will not be able to access it.

Ensure that all functionality available using a mouse is available from a keyboard. The **tab** key is used as the primary key for moving through the page content. Ensure that the order used is logical and that all items have an active state set so that there is a visual clue for where the user is on the page. The other primary keys used are the **space** and **enter** keys.

Users of screen readers use a number of keyboard shortcuts in using their applications. Take care if you are assigning keys that you are not conflicting with keyboard shortcuts from screen readers such as Jaws, NVDA and VoiceOver.

Use of applets and plug-ins

Any applet or plug-in used for content on a webpage must be accessible, or else an accessible alternative that provides the same content and functionality must be provided.

11.0 Content Standards

Unlike previous versions of this document, ESA no longer prescribes particular content standards or content standard versions; for example, there is no prescription of a particular format for images or a particular version of XML. That said, the following standards for specialist content are recommended, as they are well established in NDLRN resources.

- MathML (Mathematical Markup Language) is preferred for creating mathematical expressions within web pages, and making those expressions accessible in assistive technologies. Other standards (e.g. OpenMath) may be preferable if the semantics rather than the presentation of the mathematical expressions is a priority.

- QTI 2.1 is the preferred standard for online assessment items. Online assessment items are covered by a separate *Technical specification guide for online assessment items*, which is under development by ESA.

12.0 Digital resources and apps

Any project which includes digital resources or apps must ensure that the needs of users with disabilities are incorporated. This needs to be done from the design stage right through. No matter what the technology used, the user experience should be the primary consideration.

The accessibility features available in game applications are further advanced than browser based options such as HTML5 combined with WAI-ARIA, DOM, accessibility API's and scripting. The technology is progressing rapidly so we can only make recommendations; each project then needs to make decisions on how best to make their product accessible.

Our preferred option is to have digital resources written as HTML5 for use in a web browser. A key advance with HTML5 is the number of control elements that are now keyboard accessible (such as audio, datalist, colour, date, search to list a few); they can be made more accessible with the use of an accessibility API and WAI-ARIA. No matter what combination is decided upon, effort needs to be put into testing the accessibility available because the behaviour will not be consistent across web browsers.

If apps are produced they should be developed for iOS and Android as a minimum. The choice of game engine and scripting toolkit used will determine a lot of the accessibility features available, so that should be one of the primary criteria used in selecting the development tools.

Developers need to decide on the level of accessibility to be incorporated. The requirements to meet WCAG 2.0 level AA can now be met using the tools available, but developers need to be aware of the limitations that accessibility requirements put on the work, and allowances need to made from the start. Including accessibility features can extend the market of the resource, for example by enabling users to select a lower level of difficulty, or adjusting the time allowed might enable the resource to be used at lower year levels.

Minimum requirements for digital resources

The following should be implemented for all digital resources:

- Resources and the actions within must be keyboard accessible
- Users must be able to extend time limits where set
- Where a level of difficulty is used it must be able to be adjusted
- Non-essential alerts should be optional
- Warn users when a time limit is about to expire
- All instructions must be available in multiple formats, ie text and audio

Apps are different

Apps should be developed for Apple, Android and Windows mobile devices and can also be developed for use on OS X, Chrome and Windows desktop operating systems. It is up to the project manager to decide which systems and devices will be supported. Software development applications allow a developer to create an app once and export it to operate on desktop computers and mobile devices.

Exporting an app for multiple device types can present challenges when it comes to making the apps accessible. Unless the development application has the functionality to include accessibility information (such as labels, hints, traits and attributes) that can be used by all of the platforms on which the app is used, then it might be more efficient to develop individual versions for each platform. When selecting a development application the accessibility features should be a primary consideration.

The tools available for developing apps are evolving rapidly, and the options available should be investigated based on the requirements of each project. The range of devices on which the app is to be made available (e.g. tablets and mobile phones) and the accessibility features should be a primary consideration when making this decision: it can be an enormous amount of work to make an app accessible if the elements used do not have accessibility fields included.

12.1.1 Key requirements for apps

The following should be implemented for all apps:

- All items must have **labels** text included and **frame** information
- **Traits** information should be included where necessary and **value** information supplied where the field is a variable
- Users must be able to navigate the game using only the keyboard
- Allow users to control the size, colour and contrast of text
- Context or activation changes should occur only when the user takes a direct action on a UI element that has focus
- Audio prompts must always accompanied by a visual prompt or notification
- Users must be able to extend time limits where set
- Where a level of difficulty is used it must be able to be adjusted
- Non-essential alerts should be optional
- Warn users when a time limit is about to expire
- All instructions must be available in multiple formats, ie text and audio