Canadian Heritage Patrimoine canadien

# CANADIAN CULTURE ONLINE

## Guide to the Technical Requirements and Recommendations

**Version 4.0**
**Last updated: October 26, 2007**

***La version française de ce document est aussi disponible.***

Canadä

# Table of Contents

# Introduction

This document provides a complete description of the technical requirements that projects funded by Canadian Culture Online (CCO) **must** meet. Additionally, it includes a series of recommended practices that the designers and developers of CCO-funded projects are urged to follow.

CCO-funded projects **will be evaluated against the requirements.** Following this technical evaluation, fund recipients will be asked to revise their websites to comply with any requirement that has not been followed.

The requirements are based on industry best practices, as well as widely accepted standards developed by such organizations as the **World Wide Web Consortium (W3C)**[1] and the **International Organization for Standardization (ISO)**[2].

# Why Comply With Standards?

Web standards contribute to a guarantee of quality. If online content is worth the time, effort and cost of production, then it is worth following a standardized set of rules that will permit the greatest number of people to access it using a variety of hardware and software.

While straying from the required standards may sometimes appear easier in the short-term, it will eventually lead to increased maintenance costs as the site will need to be retrofitted in order to continue working with newer browsers and devices.

The requirements should not be viewed as a constraint, but rather as an investment that will

- make a website more fully **accessible** to all Canadians, regardless of geographic location, technology, or disability;

- make a website more **visible**, e.g., in search engine results;

- contribute to a website's long-term **viability**; and,

- facilitate **migration** of the website to newer versions of browsers and other devices, e.g., cell phones, PDAs, etc.

Web standards are designed for forward and backward compatibility. This means that a website complying with Web standards will likely continue to work in new browsers, and gracefully degrade to produce an acceptable result in older browsers.

[1] http://www.w3.org/
[2] http://www.iso.org/iso/

In addition, because a website may go through several teams of designers and developers throughout its lifetime, it is important that they are able to comprehend the (X)HTML and other code used to create the site. Web standards offer a common set of rules and terms that every developer can follow and understand, thus shortening development time and yielding pages that are easier to maintain.

# Note on Accessibility

In the 2001 Participation and Activity Limitation Survey (PALS)[3], undertaken by Statistic Canada, approximately one in eight (12.5%) Canadians were identified as having a disability. One great benefit of the Web is that, without much effort, most people with various visual, hearing, or mobility impairments can be provided direct access to content that would otherwise be unavailable to them.

In 1999, the W3C's Web Accessibility Initiative (WAI)[4] published the Web Content Accessibility Guidelines (WCAG) 1.0[5] for making Web content accessible to people with disabilities. By following these guidelines, Web developers can help ensure that disabled users, from those with mild colour vision or motor skill deficits to those that are totally blind or deaf, can access Web content. Web pages that implement the WCAG 1.0 have the added benefit of being generally more accessible and usable for everyone, both disabled and non-disabled.

While the technical requirement of CCO do not explicitly require the implementation of the WCAG 1.0, a number of CCO requirements are directly related to and effect the same outcome as some of the more important accessibility guidelines. For instance, the requirement that images [5.3b], as well as video [5.4a], audio [5.5a], and animation [5.6a] files, have text-equivalents meets checkpoint 1.1 of the WCAG 1.0[6]. The requirement that all key features of a site be available if client-side scripting is disabled or not supported reflects checkpoint 6.3[7].

Should fund recipients wish to implement the WCAG 1.0 to a fuller degree, they are encouraged to do so, though it is not required. Additionally, a set of errata and extensions to the WCAG 1.0[8] has been published by the WCAG Samurai[9], and could prove helpful in the application of various guidelines in the WCAG 1.0.

[3] http://www.statcan.ca/Daily/English/021203/d021203a.htm
[4] http://www.w3.org/WAI/
[5] http://www.w3.org/TR/WCAG10/
[6] http://www.w3.org/TR/WCAG10/#gl-provide-equivalents
[7] http://www.w3.org/TR/WCAG10/#gl-new-technologies
[8] http://wcagsamurai.org/errata/errata.html
[9] http://wcagsamurai.org/

# Canadian Culture Online (CCO) Technical Requirements and Recommendations, version 4.0

**Note:** In this document, *requirements* are **mandatory**, while *recommendations*, although optional, are **strongly recommended.** CCO-funded projects **will be evaluated against the requirements** listed below. Following this technical evaluation, fund recipients will be asked to revise their website to comply with any requirement that has not been followed.

# 1.0 Markup

## 1.1 Markup Language

### Requirement

a) All Web page content **must** be marked up using the “Strict” or “Transitional” Document Type Definition (DTD) for either HTML 4.01 or XHTML 1.0.

**Note:** Frames **must not** be used, and therefore the use of the “Frameset” DTD is **not permitted.**

**HyperText Markup Language (HTML)**[10] and **Extensible HyperText Markup Language (XHTML)**[11] are markup languages recommended by the World Wide Web Consortium (W3C) for describing the semantic structure of content, that is, identifying what is a heading, a paragraph, a list, etc., and how these different elements are related within a document.

(X)HTML comes in different versions or **Document Type Definitions (DTD).** The version of (X)HTML used is declared at the top of the Web page in what is called a **DOCTYPE** or Document Type Declaration. Of the three available DTDs for (X)HTML, that is, Strict, Transitional, and Frameset, only the first two may be used. The Frameset DTD is used for sites that implement frames, but since frames introduce serious barriers to accessibility, it is best practice to avoid them.

For example, the DOCTYPE for the **XHTML 1.0 Strict** DTD is the following:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

[10] http://www.w3.org/TR/html4/
[11] http://www.w3.org/TR/xhtml1/

### Recommendations

Fund recipients are encouraged to implement proper **semantic markup** in their Web pages. Semantic markup describes the content, as opposed to how the content is to be presented or displayed.

For example, a site menu is normally just a list of links. As such, it should be marked up as a list (e.g., `<ul>`) because the list element describes the semantic structure of that content. If the site's design calls for the menu to be displayed as a horizontal menu bar, some developers might be tempted to mark up the menu as a horizontal row in a `<table>` element, but this mixes the content's semantic structure with its presentation. More importantly, it assigns the menu the wrong semantic structure: the `<table>` element is for marking up tabular data, but the site menu is a list of links. Instead, the menu should rightly maintain its proper semantic structure as a list element, while also being displayed horizontally through the use of appropriate Cascading Style Sheets (CSS) (see 3.1 Cascading Style Sheets).

Similarly, a paragraph should be marked up as a paragraph, as opposed to, say, text separated by `<br />` elements within a generic `<div>` element. A document that includes several levels of hierarchical headings should make use of the different (X)HTML heading elements (i.e., `<h1>` to `<h6>`) in order to add the appropriate hierarchical semantic structure to the document and its sections. The heading elements can then be styled using CSS.

## 1.2 Code Validation

### Requirement

a) All (X)HTML code **must** be completely error-free and validate against the chosen W3C-published DTD for either HTML 4.01 or XHTML 1.0, whichever is used.

**Note:** The one exception to this requirement is the proprietary `<embed>` element. While the `<embed>` element **must not** appear in the raw (X)HTML document that is served to the browser, it may be dynamically added to the (X)HTML page via client-side scripting.

The W3C provides access to an **online HTML validator**[12].

As a markup language, (X)HTML has a grammar, that is, a set of explicit rules guiding how it is to be written. Validating (X)HTML code is the same as running a word processor document through a spell checker. Just as a document is free of spelling errors or not, (X)HTML code is either valid or not.

Even when Web pages look fine in a certain browser, there may still be errors in the (X)HTML code. Invalid code can jeopardize the site's accessibility in different browsers and provide incorrect information to search engine spiders.

[12] http://validator.w3.org/

While (X)HTML is a standardized language, not all Web browsers implement (X)HTML elements in the same way. One example is the `<object>` element that is used to embed in a Web page Flash movies and other multimedia content, such as Quicktime movies, or even other Web pages. Unfortunately, the implementation of the `<object>` element is not consistent across browsers, meaning that content loaded via `<object>` may not appear properly in certain browsers. However, the proprietary `<embed>` element, while not part of the official (X)HTML language, has very good cross-browser support, and will normally work where the `<object>` element fails.

The valid `<object>` element should be used in all cases, though it is now preferable, in the case of browsers with a broken `<object>` implementation, to use client-side scripting to dynamically replace the `<object>` with an `<embed>` element (see Recommendations under 5.6 Animation and Flash). Since most (X)HTML validators do not parse the Web page as modified by client-side scripts, the raw (X)HTML page will continue to validate in online validators, assuming that it has no other (X)HTML errors. However, the final page as rendered by the browser will include the invalid `<embed>` element that was added dynamically via client-side scripting. While this means, strictly speaking, that the modified page as read by the browser is not valid, the page's content is made available to browsers capable of rendering the `<embed>` element. In this case, a small concession regarding valid markup has been made in the name of browser compatibility and user access to content.

## 1.3 Character Encoding

### Requirement

a) All Web pages **must** use and declare either UTF-8 or ISO-8859-1 character encoding.

**Note:** For languages that use characters not available in UTF-8 or ISO-8859-1, see requirement 3.6b.

A character encoding is a method for translating one set of codes to another. In the case of (X)HTML pages, a character encoding provides a way for a computer to translate the set of bytes sent from a Web server into a set of written characters that the receiving Web browser can interpret.

The two most common character encodings for (X)HTML are ISO-8859-1 (also known as Latin-1), a standard character encoding for the Latin alphabet, and UTF-8, which stands for 8-bit Unicode Transformation Format, an encoding scheme for most of the world's writing systems. **Unicode**[13] is a standard encoding system that includes a unique entry for every character in almost every language.

[13] http://www.unicode.org/

# 2.0 Scripting

## 2.1 Client-side Scripts

### Requirements

a) All client-side scripts **must** be compliant with ECMAScript, version 3, also known as ECMA-262.

Client-side scripts are often incorporated into Web pages for a variety of purposes, including increasing the interactive quality of a page and extending the capabilities of the markup language. For example, a script could be created that alternates one graphic with another whenever a user drags the mouse cursor over it.

Client-side scripts are downloaded, interpreted and applied by the browser, which is what makes them "client-side," as opposed to server-side scripts written in, for example, PHP, ASP, Python, or Perl, which are applied by the **server**.

**JavaScript**[14], the most broadly used client-side scripting language, was originally developed by Netscape, and is now maintained by the Mozilla Foundation. Another scripting language, **Jscript**[15], was developed by Microsoft. Shortly after the appearance of these languages, the European Computer Manufacturers Association (ECMA) proposed a standard, known as **ECMAScript**[16]. Both the Mozilla Foundation and Microsoft have developed versions of their scripting languages that comply with this standard.

Client-side scripts can be extremely useful as they are able to access and modify the (X)HTML **Document Object Model (DOM)**[17]. The DOM is a basic, platform-independent model of all the elements or objects in an (X)HTML document, and enables client-side scripting languages like JavaScript to access and manipulate any element in the page, and even create and add new ones. This is known as DOM scripting, and is the preferred application of client-side scripts, like JavaScript, in the context of (X)HTML pages.

b) Key features of the site **must** function if client-side scripting is disabled or not supported.

Some users will have client-side scripting disabled in their browser. In some cases, it will be disabled to ensure a more secure browsing experience. Corporate firewalls also will occasionally disallow client-side scripting. Additionally, some assistive technologies are not fully compatible with client-side scripts. As such, it is an important accessibility best practice to guarantee that a site's key content and functionality be available with or without client-side scripting.

[14] http://developer.mozilla.org/en/docs/JavaScript
[15] http://msdn2.microsoft.com/en-us/library/hbxc2t98.aspx
[16] http://www.ecma-international.org/publications/standards/Ecma-262.htm
[17] http://www.w3.org/DOM/

### Recommendations

Following the progressive enhancement approach (see Recommendations under 4.1 Browser Compatibility) and the best practice of separating presentation (CSS – see 3.1 Cascading Style Sheets) from content ((X)HTML), fund recipients are strongly encouraged to also separate behaviour (JavaScript) from content in the same way. This is known as unobtrusive JavaScript, as all JavaScript is loaded from external files and the basic (X)HTML page code includes no JavaScript at all, including inline event handlers such as `onclick` or `onmouseover`.

By linking to external JavaScript files that dynamically alter the (X)HTML Document Object Model (DOM), desired modifications or additions to the Web page can be made, leaving the original (X)HTML content served to the browser untouched. Unobtrusive JavaScript is an approach that improves the maintainability of the website, since, for instance, updates to JavaScript functions associated with event handlers can be made from within the JavaScript code itself, as opposed to having to modify all of the Web pages had the JavaScript functions and event handlers been written inline in the (X)HTML source code.

CCO recommends that fund recipients develop client-side scripts using either JavaScript version 1.5 or JScript version 5.5, which most closely comply with ECMAScript, version 3.

# 3.0 Layout and Design

## 3.1 Cascading Style Sheets

### Requirements

a) Cascading Style Sheets (CSS) levels 1 and/or 2 **must** be used to establish the presentation of Web page content.

In the development of Web pages, it is best practice to separate the **presentation** of content from its **structure**. While (X)HTML is used to determine the semantic structure of page elements, **Cascading Style Sheets (CSS)**[18] is a presentational language and the preferred method for defining the visual style and layout of page elements.

Using CSS has added benefits, such as making Web page file sizes smaller, thereby reducing the time it takes a browser to load the pages. CSS also allows for greater control over the structure of the content; enables better indexing by search engine spiders; improves accessibility; and, greatly diminishes the time and energy required to redesign the site.

b) All CSS **must** be valid (i.e., completely error-free).

**Note:** The W3C provides an **online CSS validator**[19].

Like (X)HTML, CSS has a particular grammar that must be followed to ensure that browsers interpret it correctly.

c) All core website content and functionality **must** be available without CSS.

Users may override the site's specified style sheet. They may also be using older or text-only browsers, or other devices, e.g., mobile phones, that do not render CSS. In all of these cases, users must still be able to access the content.

### Recommendations

It is strongly recommended that the website's presentation be entirely separated from its content by making all style (CSS) declarations in external files that are loaded from within a page's `<head>` element using either the `<link>` or the `<style> @import` methods, or both.

Fund recipients are strongly urged to implement a print style sheet that provides a printer-friendly page layout that, for example, prevents certain elements and features (e.g., decorative images, site

[18] http://www.w3.org/Style/CSS/
[19] http://jigsaw.w3.org/css-validator/

menu, colour) extraneous to the page's content from being printed. Print style sheets can be distinguished from other style sheets that a page loads by setting the relevant `<link>` or `<style>` element's `media` attribute to "print" (i.e., `media="print"`).

## 3.2 Tables

### Requirements

a) The use of layout tables is **not permitted**.

(X)HTML tables are for marking up tabular data, that is, content that would normally be placed in a table format with meaningful row and column headers. Tables are not intended for positioning content in a grid-style layout. Using such layout tables is a misuse of the (X)HTML `<table>` element, and can cause accessibility problems.

For instance, a screen reader will read page content in the order that it appears in the (X)HTML source code, that is, left to right and row by row, thus linearizing the table's content. With layout tables, however, the reading order of the linearized table content may not be the same as the reading order indicated by the table's visual layout, and thus not make sense when read by a screen reader.

b) Data tables **must** incorporate row and/or column headers.

True data tables can be quite complex, and so steps must be taken to ensure that data tables are fully accessible. In (X)HTML markup, the `<th>` element is used to indicate table header cells, and the `<td>` element is used for table data cells, as well as for those that act as both data and header cells.

c) Data tables **must** incorporate the `scope` attribute.

The `scope` attribute identifies the group of table data cells for which the current header cell provides header information. A `scope` attribute set to "col" indicates that every cell in the relevant column is related to that column header. A `scope` attribute set to "row" indicates that every cell in the relevant row is related to that row header.

d) A descriptive name or title for each data table **must** be provided using the `<caption>` element.

## 3.3 Pop-up Windows

### Requirement

a) The use of pop-up windows is **not permitted**.

Many usability and accessibility problems arise through the use of pop-up windows, and can confuse users. For someone who is blind, there is usually no indication that a new window has opened. Also, there are often no navigational elements such as a “back” button. In addition, content in pop-up windows is often not indexed by search engines, thus neither searchable nor retrievable.

## 3.4 Image Maps

### Requirements

a) The use of server-side image maps is **not permitted**.

An image map is an image that has been assigned a set of coordinates by which different areas of the image become hyperlinked. For instance, an image of an automobile may have coordinates set for each major part of the car, enabling users to click on any of these areas and be directed to a page on that specific car part.

With server-side image maps, the determination of map coordinates, what area was clicked, and what page to return to the user, is made by the server upon receiving input from a user’s mouse click. Server-side image maps cannot be made accessible because all of the information about the different areas within the map is held by the server, and not available to the browser.

b) While CCO discourages their use, client-side image maps **must** be made accessible by including appropriate `alt` attributes for both the `<img>` and `<area>` elements used by the image map.

On the other hand, a client-side image map and its active areas are established within a page’s (X)HTML code, and so managed by the browser. Each `<area>` element of an image map can have an `alt` attribute, thus making the image map accessible. However, as image maps are graphics, they can take longer to download for those with slower connections, and if the browser cannot display images, or if images have been turned off, it is not evident what links are available in the image map.

## 3.5 Screen Resolution

### Requirement

a) The website **must** be effectively viewable (i.e., no horizontal scroll bar) at monitor resolutions of both 800x600 and 1024x768.

Different users will have their computer monitors set to different screen resolutions. It is normally ideal, except, perhaps, in the case of an extremely large computer monitor, to have a layout that stretches the display to fill the browser window at any screen resolution. However, in order that Web pages be usable without requiring horizontal scrolling on the part of the user, it is a good idea to ensure that the Web page design fits effectively within a full-screen browser window at the most common screen resolutions.

## 3.6 Font

### Requirements

a) The font-size of text **must** be established in a way that allows it to be increased and decreased using the built-in functions of popular browsers. In other words, font-size **must not** be declared in pixels (px), or in absolute units such as points (pt), but rather as relative units such as percentages (%) or “ems” (em).

The ideal font-size for screen legibility is subjective since users have different preferences and visual ability. As such, it is important to allow users to increase and decrease the text size as they wish. Fortunately, almost all browsers have this functionality built in, though to function it requires that the content’s font-size be declared using relative units (e.g., percentage (%) or “em” (em) units) that can scale, as opposed to absolute units (e.g., point (pt) units) that cannot.

Additionally, while almost all browsers will allow the resizing of text coded in pixels (px), Microsoft’s Internet Explorer **does not**, and for this reason pixels should normally be avoided when sizing text.

b) If the website employs a language that uses characters not available in the UTF-8 or ISO-8859-1 character encodings, users **must** be provided free access to download the required font that will allow them to view the content. (See also requirement 1.3a.)

Some languages have special diacritics (accents and letters) that have not been given distinct Unicode numbers. When a website uses a non-Unicode font, site visitors are not able to read the content on their computers unless they have installed on their computers the same font that was used by the website developer.

## 3.7 Visual Contrast

### Requirement

a) Web pages **must** be developed with sufficient visual contrast between text and background colours in order to be legible to all users.

**Note:** Various online tools for checking the visual contrast between text and background colours are available, for example, the **Colour Contrast Check Tool**[20].

[20] http://www.snook.ca/technical/colour_contrast/colour.html

The ability to distinguish between different colours varies by individual. To ensure that the greatest number of people can read a website's text content, it is important to use sufficiently high contrast combinations where text and background colours are concerned. The W3C has established an **algorithm**[21] that indexes the contrast ratio of any two colours, and allows different colour combinations to be graded for contrast adequacy.

[21] http://www.w3.org/TR/AERT#color-contrast

# 4.0 Browsers and Input Devices

## 4.1 Browser Compatibility

### Requirement

a) All core website content and functionality **must** be accessible in any browser, although its presentation may vary.

**Note:** While the site's presentation or behaviours may not be consistent across all browsers or devices, all users, regardless of their browser, **must** have access to the site's basic content and functionality.

With very little planning at the beginning of a Web project, it is easy to ensure that content is available through any browser. As a central purpose of CCO-funded Web projects is to make digital cultural content available to Canadians, it is important to realize that Canadians, as well as individuals worldwide, will be accessing this content through various browsers and devices.

### Recommendations

It is strongly recommended that fund recipients develop their sites following the **progressive enhancement** approach, a Web development strategy that produces, at the most basic level, a website whose core content and functionality are available to all users, no matter what browser or device is used. By progressively adding CSS presentational and JavaScript behavioural effects in a layered fashion, the site provides an increasingly enhanced experience for those with browsers that can take advantage of the added features. However, the site's core content and functionality remain available to everyone, regardless of browser or device.

For example, a timeline of major historical events might be developed using Adobe Flash, and require that users have at least version 8 of the Flash Player installed in their browser. In this case, the information in the timeline is probably not accessible to those using assistive technologies such as a screen reader, and certainly not to those using a Mac with MacOS 9.1, which can only run up to version 7 of the Flash Player. Instead, by creating a simple (X)HTML text version of the timeline and including this as the default content in the (X)HTML page, it is guaranteed that all users will be able to use the timeline. By additionally linking this page to an external JavaScript file that checks if the browser has the appropriate version of the Flash plug-in, the page's (X)HTML code could be dynamically modified by the JavaScript (assuming that the browser is capable of implementing the required JavaScript methods) to load a Flash version of the timeline. Users of suitably capable browsers would thereby receive an enhanced Flash version of the timeline, with no loss of accessibility for those using browsers that are not so capable.

## 4.2 Input Device Independence

### Requirement

a) All core website content and functionality **must** be accessible, at a minimum, to **both** mouse and keyboard input.

A good portion of users have difficulties using a mouse, and rely on different types of input device to communicate with and manipulate the browser interface. Input devices may include pointing devices, such as a mouse or trackball, as well as keyboards, braille devices, head wands, microphones, and others. Users must be able to interact with a Web page's basic content and features using the input device of their choice and according to their needs. For the most part, if a website is keyboard-accessible, it will be accessible using most other input devices, since these tend to imitate keyboard functionality.

# 5.0 Content Types and Formats

## 5.1 Plug-ins and Specialized Software

### Requirements

If a plug-in or specialized software is required to view, listen to, or use certain content (e.g., audio, video, animation, text):

a) The plug-in or software **must** be freely available to users and platform-independent.

b) A link to a page that allows users to download the plug-in or software **must** be provided on those pages where the plug-in or software is required, and **before** the content is presented.

c) A sentence alerting users that a plug-in or specialized software is required to access the content **must** appear on those pages where the plug-in or software is needed, and **before** the content is presented.

The Web has made it increasingly easy to serve multimedia content, including video, audio, and animation files. Usually, such content cannot be natively displayed by a Web browser, and additional plug-ins or software needs to be installed on a user's computer. As a result, it is best practice to provide multimedia content in formats that are technologically available to all users, regardless of computer platform. Because users may be required to download and install extra software, in addition to downloading the multimedia content itself, it is also best practice to alert users of the format of the multimedia files, and provide access to the source of the required software.

## 5.2 Text (See also 5.1 Plug-ins and specialized software)

### Requirement

a) All text-based content **must** be displayed primarily in (X)HTML format.

**Note:** Text-based content can **additionally** be developed in a proprietary format, for viewing or printing, with the use of a freely available and platform-independent plug-in or software (e.g., **Adobe's PDF**[22], which requires the use of the Adobe PDF viewer, **Adobe Reader**[23]).

Text-based content can **also** be delivered in formats such as RTF or ASCII, or as a delimited text file. Content produced in such formats is acceptable if the files are intended for download, or for storage

[22] http://www.adobe.com/products/acrobat/adobepdf.html
[23] http://www.adobe.com/products/acrobat/readstep2.html

or manipulation by users outside of the browser environment, and if they are not meant to be a substitute for content created in (X)HTML.

### Recommendations

To promote the long-term viability of text-based content, CCO recommends that fund recipients create and manage content in a structured format suitable for delivery as (X)HTML files. In most cases, storing text-based content as (X)HTML, XML or ASCII text will be the most appropriate option. This will reduce the dependency on any content creation or management software and on the server environment where it is stored.

Text-based content can also be developed and managed in a database or content management system (CMS). CCO recommends that fund recipients choose a system that will allow the content to be exported in a standardized format. A CMS must produce valid (X)HTML pages. This will ensure that the content managed in such environments is not completely dependent on a single application or platform and, at a minimum, that it can be migrated to new environments with greater ease.

## 5.3 Still Images

### Requirements

a) JPEG, GIF, or PNG formats **must** be used for photographs and graphics in Web pages.

**Note:** CCO may grant an exemption to the above requirement in certain situations where the use of proprietary solutions is needed to meet project objectives. This could be the case, for example, for geographic maps. In such cases, specialized solutions could be chosen to provide users with greater flexibility in viewing the digitized objects. In order to obtain an exemption, fund recipients **must** request permission from CCO at the start of the project, with a justification for the use of such solutions.

Most graphical browsers and operating systems have built-in functionality for viewing images in JPEG, GIF, or PNG formats.

b) All images, including graphical representations of text (including symbols), images used as list bullets, spacers, and graphical buttons, **must** be accompanied by text-equivalents in the form of `alt` attributes.

The (X)HTML `alt` attribute is used to define the "alternate text" for an image. It tells the reader what she is missing on a page if she cannot, for whatever reasons, view the images. The browser will then display the `alt` attribute’s content instead of the image. For those using assistive technologies such as a screen reader, the `alt` text will be read to the user. Including `alt` attributes has the added benefit of allowing the images to be indexed by search engines, thus enabling their retrieval.

c) All large images and large graphic files **must** first be presented to users through thumbnail or other smaller image previews. The thumbnail or image preview **must** be hyperlinked so that when users click on it, they are directed to the larger version of the same image.

To prevent users, especially those with slower Internet connections, from having to download the full-size versions of images in order to see if the images are of interest, it is best practice to provide users with thumbnails or smaller image previews of the full-size images. This will allow users to decide whether or not they want to view a full-size image before having to download it.

d) Original, high-resolution (e.g., TIFF or PNG) versions of images used in the website **must** be properly stored and backed-up in an environment such that fund recipients can access them at a future date.

The original versions of image files must be properly stored, backed up, and migrated according to appropriate long-term preservation practices so that fund recipients can access them at a later date, for example, in the event of data loss or a technical audit.

Digital preservation is specifically concerned with the preservation of electronic (digital) content. It is defined as "a broad range of activities designed to extend the usable life of machine-readable computer files and protect them from media failure, physical loss, and obsolescence."[24] Digital preservation activities are divided "into those that promote the long-term maintenance of a bitstream (the zeros and ones) and those that provide continued access to its content."[25]

By applying digitization and preservation best practices, and conforming to CCO technical requirements, fund recipients maximize their chances of keeping their websites—and most importantly, their contents—accessible and findable over the long term, despite organizational and technological changes.

For advice on digitization and preservation practices, contact CCO (see Contact Information at the end of this document).

### Recommendations

`Alt` attributes are intended for those users who, for whatever reasons, cannot view the images in a Web page. Following requirement 5.3b, every `<img>` element must include an appropriate `alt` attribute, though what counts as appropriate depends on the image and its use or function in the page, from a user's perspective. An image's `alt` attribute, like any text-equivalent for video or audio, should be understood as a replacement for the image. When determining an image's `alt` attribute content, ask yourself how you would describe that image, in its specific context, over the telephone or in an email to someone who cannot see it. Remember that most users who cannot view images will usually either see the `alt` attributes' content as text, or hear it in the case that they are using a screen reader.

[24] Cornell University. Digital Preservation Management Tutorial. October 2005. http://www.library.cornell.edu/iris/tutorial/dpm/index.html.
[25] *Ibid*.

For example, an image of a map showing directions from one location to another should have an `alt` attribute that textually reproduces the directions indicated visually in the image. The `alt` attribute for an image of a sculpture should describe the image in a way that captures the intended meaning of the image as it is used in the page, perhaps as an example of a particular sculpting technique.

On the other hand, many images are purely presentational, such as stock photos in corporate banners, or custom images for list item bullets. In most cases, these images provide no meaningful content or information to users, and are simply there for presentational effect. While, as images, they require `alt` attributes, these `alt` attributes should be empty or null, i.e., `alt=””`. This way, those who cannot view the images are not asked to pay them any notice, since there is no `alt` attribute content to direct their attention. Similarly, while blank images (e.g., “spacer gifs”) included in (X)HTML content to help position elements on the page should in fact be replaced by proper CSS positioning, such images should equally take an empty `alt` attribute as they do not impart any meaningful content or information.

Sometimes, a content image will be accompanied by (X)HTML text that serves as a description of the image. If this text, through both its position in the source code as well as on the page, is clearly associated with the image, there is usually no need to repeat it in the image’s `alt` attribute. Unless there is meaningful information in or about the image that is not found in its description, and which could rightly be added to the image’s `alt` attribute, the `alt` attribute should remain empty, since the description will act as the image’s text-equivalent. Often, though, developers, in a best effort to provide `alt` attributes, will reuse the image’s textual description for this purpose. Unfortunately, this causes needless repetition for those users who, relying on `alt` attributes, will see and/or hear the same text twice, once from the `<img>` element’s `alt` attribute, and once from the image’s regular text description on the page.

`Alt` attributes should not be longer than 100 characters, spaces included. Should an image’s `alt` attribute be longer than 100 chartacters, it is better to provide the complete text-equivalent as (X)HTML text next to the image, while giving the image itself an empty `alt` attribute.

## 5.4 Video (See also 5.1 Plug-ins and specialized software)

### Requirements

a) All video files **must** be accompanied by (X)HTML text-based summaries or transcripts that allow any user to understand the essential content of the individual video clips without having to view them.

As some users are unable to see and/or hear, the **meaningful** content of a video file must be made accessible in a fashion that allows users to understand it even if they cannot view or hear the video file’s content.

A blind person may well wish to listen to a video, relying on a text description of its visual content. A deaf person may want to watch a video, while relying on text captions or a time-indexed transcript of any dialogue or audio the video contains. Including such text-equivalents is crucial to accessibility, and has the added benefit of making the video content indexable by search engine spiders, which can generally only read text.

b) Video files that are loaded into a browser-embedded player **must not** start automatically, and the player **must** include controls for starting and stopping the video.

To prevent large video files from being automatically streamed without the permission or control of the user, especially in the case of slow or dial-up Internet connections, and to avoid surprising users with unexpected video and/or audio content, it is best practice to not have video files begin playing automatically, and to include controls that allow users to start and stop the video as they wish.

c) All links to large video files (i.e., files greater than 50 Kb) **must** be accompanied by a label indicating the file's size and duration.

To prevent users, especially those with slower Internet connections, from having to download a large video file that they might not otherwise, were they aware of how large the file is, it is best practice to label the link to the video clip with the file's size. This way users can make an informed decision prior to downloading the video file.

d) Where a codec is used to produce video files, it **must** be included in a common platform (such as the Apple, Real or Microsoft platforms) or freely-available for installation by users. A link to the codec **must** be provided for users who need to download and install it.

A video codec is software that both en**cod**es and **dec**odes (and/or **co**mpresses and **dec**ompresses) a video file. A codec enables, for instance, a raw video file to be encoded in a certain format that compresses the file's size, making it easier to send over the Internet. The same codec equally enables the recipient of the video file to decode and decompress it in order to play it back. H.264 is an example of a recent codec that produces good quality video files at decent compression ratios.

e) If video files are prepared for delivery in a high-bandwidth environment, a low-bandwidth version **must** also be prepared and provided to users.

**Note:** Some streaming systems allow producers to prepare a single video file that can be played at various streaming speeds corresponding to multiple Internet access speeds. In cases where such systems are used, a single version of a video file is sufficient.

Video files can be delivered as streaming video, in which case each individual portion of the video file that is sent by the Web server and received by the user's computer is immediately displayed and then discarded, leaving no video file to be saved on the computer.

Video files can also be served such that they must be entirely downloaded and saved on the user's computer prior to playback.

A video file can additionally be delivered as a "progressive download." In this case, while the video file is actually downloaded and stored on the user's computer, it will begin playing as soon as enough of the file's content has been downloaded (i.e., buffered), which means that the user does not have to wait for the entire file to be downloaded.

Video files must be prepared specifically for the particular delivery method selected, that is, streaming, download, or progressive download.

## 5.5 Audio (See also 5.1 Plug-ins and specialized software)

### Requirements

a) All audio files **must** be accompanied by (X)HTML text-based summaries or transcripts that allow any user to understand the essential content of the individual audio clips without having to listen to them.

Some users are unable to hear, and so the **meaningful** content of an audio file must be made accessible in a fashion that allows users to understand it even if they cannot hear it. A deaf person will need to rely on text captions or a time-indexed transcript of any dialogue or narration the audio contains. Including such text-equivalents is crucial to accessibility, and has the added benefit of making the audio content indexable by search engine spiders, which can generally only read text.

b) Audio files that are loaded into a browser-embedded player **must not** start automatically, and the player **must** include controls for starting and stopping the audio.

To prevent large audio files from being automatically streamed without the permission or control of the user, especially in the case of slow or dial-up Internet connections, and to avoid surprising users with unexpected audio content, it is best practice to not have audio files begin playing automatically, and to include controls that allow users to start and stop the audio as they wish.

c) All links to large audio files (i.e., files greater than 50 Kb) **must** be accompanied by a label indicating the file's size and duration.

To prevent users, especially those with slower Internet connections, from having to download a large audio file that they might not otherwise, were they aware of how large the file is, it is best practice to label the link to the audio clip with the file's size. This way users can make an informed decision prior to downloading the audio file.

d) Where a codec is used to produce audio files, it **must** be included in a common platform (such as the Apple, Real or Microsoft platforms) or freely-available for installation by users. A link to the codec **must** be provided for users who need to download and install it.

An audio codec is just like a video codec (see section 5.4 Video for a definition of video codec), except that it is designed to work with audio files. MP3 is the most ubiquitous audio codec currently in use.

e) If audio files are prepared for delivery in a high-bandwidth environment, an alternate low-bandwidth version **must** also be prepared and provided to users.

**Note:** Some streaming systems allow producers to prepare a single audio file than can be played at various streaming speeds corresponding to multiple Internet access speeds. In cases where such systems are used, a single version of an audio file is sufficient.

Audio files can be delivered in the same ways as video files (see section 5.4 Video for a description of video delivery methods), that is, as streaming files, downloads, or progressive downloads.

## 5.6 Animation and Flash (See also 5.1 Plug-ins and specialized software)

### Requirements

a) All animation (e.g., Flash) files **must** be accompanied by (X)HTML text-based summaries or versions that allow any user to understand the essential content of the individual animation files without having to view them.

**Note:** Interactive features such as slide shows or photo galleries **must** be provided through standard (X)HTML interfaces, to which an animated or Flash interface is an alternate version.

When it is not possible to produce a standard (X)HTML version of a Flash animation, such as is the case for most interactive games, an (X)HTML text-based summary describing the function and purpose of the game will be acceptable.

Standards bodies as well as industry organizations have provided both open and proprietary solutions for developing animations and animated interfaces to Web content. While several standards are available for creating animations, such as the W3C's **Scalable Vector Graphics (SVG)**[26] and **Synchronized Multimedia Integration Language (SMIL)**[27], **Adobe Flash**[28] has become the *de facto* standard for creating and using Web animations and rich interfaces.

As some users are unable to see, hear, or interact with a mouse, the meaningful content that an animation presents must be made accessible in a fashion that allows users to understand it, even if they cannot see, hear, or interact with it. Animations, especially Flash animations, are usually not keyboard-accessible, and therefore, people that cannot use a mouse cannot interact with the animation. If the animation can be made keyboard-accessible, this should be done; otherwise, such individuals will need to rely on the text-equivalent version of the animation’s content.

[26] http://www.w3.org/Graphics/SVG/
[27] http://www.w3.org/AudioVideo/
[28] http://www.adobe.com/products/flash/

b) All links to large animation files (i.e., files greater than 50 Kb) **must** be accompanied by a label indicating the file's size.

To prevent users, especially those with slower Internet connections, from having to download a large animation or Flash file that they might not otherwise, were they aware of how large the file is, it is best practice to label the link to the animation with the file's size. This way users can make an informed decision prior to downloading the animation file.

### Recommendations

It is strongly recommended that Flash files be added to Web pages using a client-side scripting solution such as UFO[29], or SWFObject 2.0[30]. For one, this will help the site where its progressive enhancement (see Recommendations under 4.1 Browser Compatibility) is concerned, as a Flash movie can dynamically replace the default and accessible version of the same content already in the page if the browser is appropriately JavaScript-enbabled and Flash-capable. Otherwise, the user will remain with the accessible and default version of the content. Secondly, such a solution serves to overcome the Microsoft ActiveX feature in Internet Explorer 6.x and above that forces users to first click on a Flash movie before interacting with it. (See also the note under requirement 1.2a.)

Fund recipients should be aware that a Web page presenting only Flash content contains no search engine indexable text. Therefore, content presented solely in Flash will not be indexed by search engine spiders. Additionally, since the majority of Flash content cannot be read by screen readers, the use of Flash involves rather serious accessibility issues that should inform any decision to implement Flash content.

[29] http://www.bobbyvandersluis.com/ufo/
[30] http://code.google.com/p/swfobject/

# 6.0 Metadata

## 6.1 Dublin Core Metadata

### Requirements

a) The following six mandatory Dublin Core elements **must** be applied to describe the most important pages of the website, such as the home page, each section's main page, and pages featuring resources for which there is sufficient context and meaning and are also worth listing in a search engine:

- Title
- Creator
- Subject
- Date Created
- Language (where applicable)
- Identifier

b) Metadata **must** be embedded in the `<head>` section of the (X)HTML page.

c) Each page described **must** feature a unique set of metadata content.

**Note:** Copying and pasting the same metadata content is bad practice as it makes pages compete among themselves to be retrieved. At the very least, no two pages should have the same identifier, title and list of subjects.

d) Metadata content **must** be in the language of the page.

**Note:** This means that a page in English **must** include metadata content in English, and a page in French, metadata content in French. The syntax (e.g., `<meta name="dc.title" content="" />` is never translated. In cases where a page is bilingual or multilingual, metadata elements Creator, Title, Subject and Language **must** be repeated to reflect all languages; however, in such cases, it is not needed to repeat the Date Created and Identifier elements.

One of CCO's objectives is to facilitate the discovery and increase the visibility of Canadian cultural content on the Internet. That is why, for projects funded by CCO, metadata is required. Metadata, or "data about data," is defined as structured descriptions of objects in real or online environments. In the digital realm, metadata, like a library catalogue, provides some basic information about digital resources. Users can search metadata records to find digital resources, much like they would search a library catalogue to find a particular book. Without structured metadata, users are left sifting through mounds of irrelevant information to find what they are looking for. Applying metadata can, among other things, help manage content, improve site navigation and manage rights associated with content.

CCO requires the application of six metadata elements for all CCO-funded projects. Other metadata elements can be added, but that decision is at the discretion of individual organizations. The application of these mandatory six metadata elements will help to ensure that sites funded by CCO can be found by Canadians and others interested in Canadian culture.

## Recommendations

To facilitate compliance with the above requirements, the template below can be directly embedded in the (X)HTML code, within the `<head>` element. Fund recipients simply have to fill it out with the appropriate metadata content that best describes the given Web page.

```
<link rel="schema.dc" href="http://purl.org/dc/elements/1.1/" />
<link rel="schema.dcterms" href="http://purl.org/dc/terms/" />
<meta name="dc.title" content="insert the title of the Web page
being described" />
<meta name="dc.creator" content="insert the name of the Web
page’s creator" />
<meta name="dc.subject" content="insert keywords describing the
Web page, separated by comma" />
<meta name="dcterms.created" content="insert the date of
creation of the Web page being described, using the YYYY-MM-DD
format" />
<meta name="dc.language" content="insert the language name or
code for the Web page being described" />
<meta name="dc.identifier" content="an identifier can be the URL
of the specific Web page being described" />
```

Note: The template above is provided for XHTML. In HTML, the trailing slash ("/") is not required before the closing bracket (">"), as in this example:

```
<meta name="dc.creator" content="Name of Web page creator">
```

# 7.0 Databases

## 7.1 Database Interoperability

### Requirement

a) If the website employs a database, the database **must** support open standards such as SQL and XML. The database solution **must** allow fund recipients to import and export the data into a range of commonly used formats that are independent of any particular platform and software rather than dependent on a proprietary one.

Many websites make use of databases to store content. Users never interact directly with the database, but do so through a browser, which will show the results of a database query.

The database requirement above is aimed at ensuring the interoperability and preservation of data.

### Recommendations

It is strongly recommended that fund recipients document their database designs and structures. Documentation about the design, the rationale behind it, and decisions behind normalization, etc., will serve several purposes. These include facilitating possible migrations, data recovery and preservation, not to mention allowing others to better understand the database itself in cases of future maintenance and software updates.

Fund recipients building databases to store large amounts of metadata are encouraged to follow the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)[31]. "Open" means that the specifications are freely available. "Archives" refers to repositories. As the OAI mission statement indicates, "the Open Archives Initiative develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content." The goal is essentially to enable access to Web content through interoperable repositories for metadata sharing, publishing and archiving.

The Protocol for Metadata Harvesting is a mechanism for interoperability between repositories. It gives data providers (for example, CCO fund recipients) technical options to make their metadata available to others, based on the open standards, HTTP (Hypertext Transfer Protocol) and XML (Extensible Markup Language). Metadata harvesters, such as cultural information portals and metadata repositories, can then harvest those metadata. Access to content is ensured through the metadata. In the OAI-PMH, the basic level of interoperability is provided by the Dublin Core metadata standard (either by using or mapping to it).

For more information, fund recipients can consult the Open Archives Initiative website[32].

[31] http://www.openarchives.org/pmh/

[32] http://www.openarchives.org/

# Contact Information

For further information concerning the technical requirements presented in this document, please communicate directly with your CCO program officer, or contact CCO using one of the methods below.

Toll-free line: 1-866-900-0001

Email: ccop-pcce@pch.gc.ca

Postal mail:
Canadian Culture Online
Canadian Heritage
25 Eddy Street, 3rd Floor (25-3-V)
Gatineau, QC, K1A 0M5