

# Creating Accessible Online Content Using Microsoft Word

Eoin Campbell, XML Workshop Ltd.

<http://www.xmlw.ie/>

People with disabilities probably benefit more than any other group from the delivery of educational materials online. Good-quality accessible course material, which is readable, navigable and usable, improves the online experience for everyone.

However, creating accessible content is not as easy as it should be, particularly with the common HTML authoring tools in widespread use today, such as Dreamweaver and FrontPage. This talk describes a process for creating content using Microsoft Word, and converting it through the medium of XML into high-quality HTML that complies with the W3C Web Accessibility Initiative Guidelines Level 2. Apart from creating accessible content, using Word as the basic authoring tool obviously has huge benefits in ease-of-use and ease of maintainability, particularly for those authors who create content for online delivery occasionally rather than regularly.

Portsmouth University (cf. <http://www.portsmouthonlinecourses.com/>) have already developed 3 complete online MSc. courses in this way, and the approach can also be used to create interactive tests in Word, suitable for import into BlackBoard and WebCT.

## Introduction

Creating high quality online learning material is hard, and attempting to address the needs of students with disabilities at the same time is harder still. This paper describes one approach to this problem, which allows you to focus on content, while using technology to look after the details of formatting the material so that people with disabilities can access it with the same ease as able-bodied students.

In addition to describing the general approach to creating accessible content from Word, this paper looks at a number of issues particularly relevant to online learning material (interactions, integration with LMSs, specialised markup languages, etc.). A more detailed technical discussion of a Word to HTML conversion system developed for general use is available online at [CAMP2003].

There are many different ways to create HTML pages. Some require knowledge of HTML markup and others don't. Approaches that require HTML knowledge include the following.

- Plain-text editors such as Notepad, vi, or EditPlus.
- HTML editors such as Dreamweaver or FrontPage.
- Through-The-Web (TTW) forms such as eWebEditPro.
- Conversion from word-processing applications such as Word, WordPerfect etc.

In general, to create accessible HTML markup requires detailed HTML knowledge and a plain-text or HTML editor. Through-the-Web and conversion approaches generally yield very poor quality markup. Microsoft Word can save documents as HTML, but the markup quality is truly appalling. There are many 3<sup>rd</sup>-party Word to HTML converters available, but any I have examined seems to focus on enabling documents to be split into multiple HTML pages, and on adding a common HTML template. I found none which attempted to generate accessible HTML.

FrontPage achieves the remarkable feat of generating very poor quality HTML markup, even though it is a HTML-aware editor.

To be fair, FrontPage 2002 (XP) does include better support for accessible HTML, but it still generates poor HTML by default.

In an ideal world, authors would be able to create and maintain content using their normal word-processor, using all the normal features one expects from a modern editor (good spell-checking, easy printing, powerful search and replace, cutting and pasting, etc.). When it is time to publish the material to a website, one should be able to save it as accessible HTML, using an appropriate HTML template so it has the required graphic design and navigation bars found on a typical web page, and publish it easily to its final destination on a website.

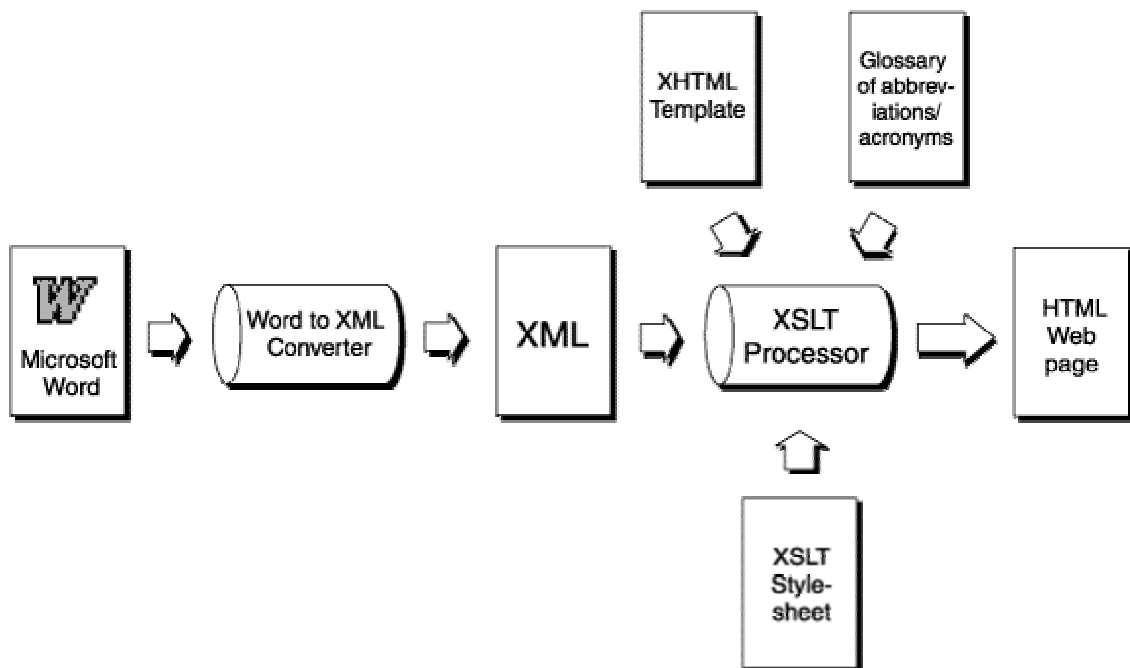
Is such a system possible? Yes, in fact there are already a number of commercial tools available which offer this basic functionality. The following products all support creating ready-to-publish web pages direct from Microsoft Word.

- YAWC Pro (<http://www.yawcpro.com/>) and YAWC Online (<http://www.yawconline.com/>)
- eXportXML (<http://www.schultz.dk/exportxml/>)
- Microsoft Word 2003, (<http://www.microsoft.com/office/preview/>)

That's the good news. The bad news is that varying amounts of customisation is required to make these products generate high quality pages for your website. In addition, knowledge of what accessibility entails is required to really maximise the quality of the output HTML.

## General system architecture

All of the above tools use the same general approach, as shown in Figure 1.



**Figure 1 Word to XML to HTML conversion process**

The authoring process is as follows.

1. Create and edit content in Microsoft Word, making use of Words' built-in styles, such as headings, lists, tables, etc.
2. Convert document from Word into XML format.
3. Convert XML document into HTML markup, and add generic HTML template to achieve the graphic design required for your target website.
4. Publish the final HTML page on your site, using FTP, WebDAV, etc.

The key to successful conversion is to convert the Word content into XML first, and then into HTML, and each of the tools listed does this. The advantage of XML (eXtensible Markup Language) is that content encoded in this format can be easily manipulated using a variety of programming languages. I will assume you have heard of XML, but you may not have heard of XSLT (eXtensible Stylesheet Language: Transformation), which has become the most popular language for manipulating XML content. XSLT, like XML, is an open standard defined by the W3C (World-Wide Web Consortium), and is widely supported in software. XSLT is built into Microsoft Internet Explorer 5, Netscape Navigator 7, and the Apache and IIS web servers, as well as relational databases like Oracle, SQL Server and IBM DB2.

From an accessibility perspective, using XML as an intermediate format, XSLT as the programming language to convert to HTML, offers the following features, all required to meet the W3C Web Accessibility Initiative (WAI) Guidelines, the *de facto* world standard for accessibility compliance.

- The HTML markup can be made to validate against a formal grammar, preferably the HTML 4.01 Strict DTD.
- Metadata can be added, preferably according to the Dublin Core Metadata Element Set, itself the *de facto* standard for HTML metadata.
- Documents can be split into multiple pages at natural breaks, such as headings.
- Inaccessible markup tags such as `<font size="+1">`, can be replaced with valid equivalents.
- Acronyms and abbreviations can be looked up from a table, and the spelled out words added to the markup.
- Context-specific information, such as a navigation aid like a 'breadcrumb' trail, can be added if sufficient metadata is available (e.g. a directory path).

Apart from the generally useful features listed above, there are some benefits of particular value to the creation of online learning materials.

- More complex HTML structures, such as drop-down lists in form fields, can be created from simpler Word structures such as tables. This means that online questions, important to force student interaction with the material, can be easily created and maintained in Word, without the expert HTML knowledge normally required.
- Alternative outputs to HTML can be easily generated. Examples include XML-encoded industry standard IMS/SCORM learning object manifests, and LMS -specific markup like the WebCT or BlackBoard online assessments (text and XML formats, respectively).

## Accessibility

Website accessibility is a complex area, and involves three major components.

- **Visual information:** encompassed in the graphic design and information architecture. This is properly the field of graphic designers and information architects, and relates to a website in general, rather than any pages in particular. It is outside the control of authors.
- **Non-visual information:** encompassed in the markup used to present information. This may be under the direct control of authors if they choose, by using a plain-text or HTML-aware editor. It is more

commonly influenced indirectly, through a dependence on the markup generated by the authoring tool, either chosen by or forced on the author.

- **Document information:** encompassed in the language, words, structures and content organisation used to convey information. This is the area where the author has complete control.

Accessibility is commonly considered to be solely a means of making websites readable by people who are blind, but this is a very narrow view. Only a very small proportion of your audience is likely to be blind. Most people with disabilities are affected by mundane, everyday ailments that affect their ability to interact with websites in some obvious or non-obvious way. For example, in the visual arena, many people are colour-blind, short- or long-sighted, and this can cause difficulties. Physically, someone with arthritis (many older people), or an injured arm or finger (sports enthusiasts), may find using a mouse or keyboard especially difficult, so suitable alternative navigation mechanisms must be defined. My 84-year-old father finds it difficult to control the mouse, so websites that use pop-up menus are impossible for him to navigate.

From an authoring perspective, there is little you can do to mitigate these particular difficulties, but there are many other areas, directly within your control, that also affect the overall accessibility of website content. Examples include spelling and grammar, the vocabulary and type of language you use, the markup of content using headings, lists, tables, etc. The editing tool used can either greatly help or greatly hinder authors in creating accessible content, so the choice of tool is important, as is an awareness of what is required to achieve accessibility.

The WAI Accessibility Guidelines (<http://www.w3.org/WAI/>) are somewhat complex to understand if you are not familiar with HTML, but nonetheless many of the requirements are not specific to HTML markup, and can equally be read as applying to content in Word. For example, the most famous accessibility requirement is that all images should use the alt tag (e.g. ``). In Word, this translates to including a formal caption after each picture, using the built-in Caption style. You can configure Word to simplify the addition of appropriate markup, by using keyboard shortcuts, menus and toolbars, and this is discussed in more detail below.

## Authoring

It is not enough to convert Word content into XML and then HTML. Unless the original Word document has been carefully marked up using the available built-in styles, the quality of the output will still be poor. Here are some rules for content markup accessibility that are directly under the control of the author.

- Use heading styles to convey document structure. Use `<Ctrl>+<Alt>+<RightArrow>` to apply a style, don't just apply a larger, bolder font style.
- Use the built-in list styles to mark up lists (e.g. List Number, List Bullet). This can be difficult, because the default Word formatting toolbar icon simply applies presentation formatting.
- Identify the natural language of the text, including any changes in language. This is easy in Word (using `Tools>Language>Set Language...`), and has the added benefit of improving the spell-checking function.
- Use tables to mark up tabular data, and mark up row and column headings. Again, this is easy, because Word has a very good table editor.

These rules all affect the quality, accessibility and even size of markup generated. Here is the same structural element, a heading, marked up in two different ways.

Presentation: `<p><b><font face="Arial" size="16pt">Arial, 16pt, bold</font></b></p>`

Structure: `<h1>Heading 1 style</h1>`

Not only is the structural markup more accessible, it also is smaller and faster, and works across a wider range of browsers. It is worth noting that specifying the language of all text is particularly important from an accessibility perspective, because screen readers such as JAWS try and pronounce words in the default

language. In an online course for French, all French text should be clearly denoted as such, or the screen reader will attempt to pronounce it as English. Even if the screen reader does not support the language, it can spell it out instead, once it knows it is different.

The conversion process depends on authors using structural markup in Word, if structural HTML markup is to be output at the other end. Therefore the author needs assistance and support to create structured content. It is quite easy to develop your own templates that provide menus, toolbars and shortcut keys to simplify marking up documents.

The following keyboard shortcuts can be assigned to insert common Word styles. Making these bindings explicit and visible by providing a toolbar and menu as well, greatly improves the ease-of-use for authors.

Table 1: Keyboard shortcuts for common styles

To insert the Word style...	Use the keyboard shortcut...
Title	<Ctrl>+T
Heading 1	<Ctrl>+1
Heading 2	<Ctrl>+2
Heading 3	<Ctrl>+3
List Bullet	<Ctrl>+8
List Number	<Ctrl>+9
Normal	<Ctrl>+0

A menu can also be used to provide the same functions, and is useful for occasional users who don't remember the shortcuts, and as a reminder for more frequent users too.

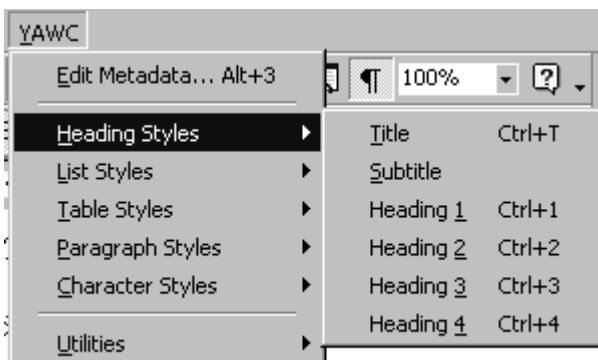


Figure 2 Word menu with explicit styling commands

The styles listed above are all built into Word, so there is no need to create them. There are a number of commonly occurring markup constructs in HTML that have no equivalent in Word, and these should be added as user-defined styles. The paragraph styles Table Title and Table Summary are essential for accessibility (mapping to the `caption` element and `table summary` attribute).

## Images

There are two common image formats on the web, GIF for graphics, and JPEG for photographs. If you use older versions of Windows (e.g. Windows 98) or Word (e.g. Word 97), then you won't have any means of creating images in these formats, unless you install 3<sup>rd</sup>-party tools. However, most Word to XML converters will automatically convert native Windows bitmap images into GIF or JPEG, which can be a great convenience. Some converters also support diagrams drawn using Words' Picture editor, thus

removing the need for any 3<sup>rd</sup>-party software, and allowing authors to maintain all their content within the Word document.

When inserting images, authors must be encouraged or forced to fill in a caption, so that when the page is converted, the resulting `img alt` attribute can be automatically filled in with sensible text. Word 2000 can be configured to automatically add a Caption style immediately after an image is inserted, to prompt the author to fill in text.

## **Equations**

Many online courses in engineering, science and maths include equations. Word has a built-in Equation Editor suitable for simpler equations, and the MathType plug-in is available for more complex equations. Therefore authors of this type of content are quite well served by the Word environment. Most converters convert equations into images, which can be included in the resulting HTML output.

Using images for equations is not ideal, both from an accessibility perspective, and if you want to allow students to download and use equations you provide. The new standard for online equations is another W3C specification, MathML, the Mathematics Markup Language. MathML is supported natively in browsers such as Mozilla, and via a plug-in for Internet Explorer, so there is now possible to use it in online course content.

The MathType Equation Editor plug-in for Word can import and export MathML (as well as other popular equation formats like TeX), so all the components are in place to be able to create and publish equations. Unfortunately doing this in a completely automated way is not yet possible, so authors must still do some manual fiddling of the HTML output if they want embedded MathML. However, we are currently working to resolve this problem for our YAWC products.

## **Metadata**

The use of metadata is a requirement for accessibility, and the standard for web page metadata is the Dublin Core Metadata Element Set (<http://www.dublincore.org/>). Metadata doesn't improve accessibility directly, but it does improve findability, as search engines can index the metadata on a site, enabling information to be found more easily.

In the context of online learning material on an intranet server, standard Dublin Core metadata may not seem very useful, but intelligent use of Subject Descriptor, Type and Keyword fields may help considerably in content administration, monitoring of what types of material students are accessing. For example, it would be useful to know which type of content is most accessed by students: case studies, interactions, lecturers' or 3<sup>rd</sup>-party course content, etc.

It is quite easy to create a dialog box using Words VBA editor to enable authors to fill in metadata fields. Many of the fields are fixed for a particular website or course, and others can be automatically deduced from the document itself. For example, the DC.Title field can be extracted from the document title, and the DC.Date.modified field from the date of conversion. Even the value of the DC.Identifier field can be at least partially if not fully deduced by the conversion software. The figure below shows how a metadata dialog box can look.

Figure 3 Dublin Core Metadata dialog box

### Interactions

Interactions to engage attention and assess the students' progress are one of the most important aspects of creating content for online courses. Unlike textual material, which can be published in the preferred format of the author, whether HTML, Word or PowerPoint, online interactions must be formatted according to the requirements of the LMS or content server. 3<sup>rd</sup>-party tools such as QuestionMark support various LMS assessment formats. However, online course interactions have a strong need for intelligent feedback to students, which is not supported by many standard tools or editors, and customisations are therefore required.

A good approach to this problem is to use pre-defined table structures in Word to create interactions, and convert these to the appropriate publication format as part of the XSLT post-processing phase. This removes a lot of complexity from the authoring process, and again enables all content to be created and maintained in a single format. Table 2 shows a multi-choice question template.

Table 2: A multi-choice question template

Type	Source	Question
MC	A. Teacher	In Spain, where does the rain fall?
Status	Answer	Feedback
Correct	The plain	Yes, the rain in Spain stays mainly in the plain
Incorrect	The mountains	No, in Spain the mountains are generally dry
Incorrect	The coast	No, the coastal areas are quite arid
Incorrect	The south	No, the south is the driest part of Spain

Attractive as this approach is, it involves a very significant amount of development to make it work well.

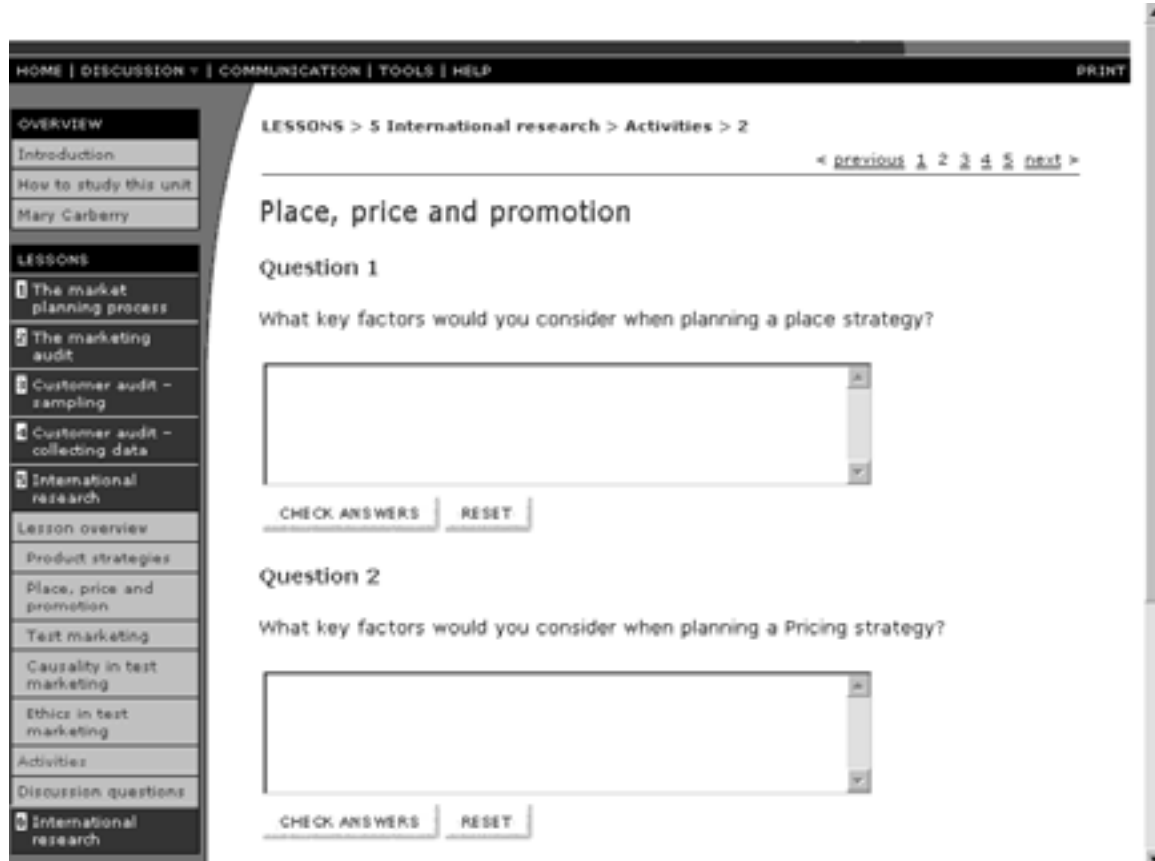
## Tool selection

There are many Word to XML converters available, and we maintain a reasonably complete list on our website at <http://www.xmlw.ie/aboutxml/word2xml.htm>. There are a number of different categories of product, so choosing the right tool for your needs isn't always straightforward. The following are issues to consider in choosing the tool.

- **Platform and version requirements:** Do Windows and Macintosh clients need to be supported? What Word version is used in your organisation? Some tools are Windows only, or require Word 2000. Others require a Java Virtual Machine.
- **Integrated with Word or stand-alone:** Some converters are available as additional menus within Word, while others require starting up a separate application. Some standalone tools convert from RTF rather than native Word format, adding an extra step to the process.
- **Software or service based:** Although most tools are installed on the desktop, there are now some online services for Word to XML and/or HTML conversion. This removes the need to install software locally, and can greatly reduce licensing, installation, configuration, and ongoing support costs.
- **Customisation effort:** Different tools generate different XML markup by default, and the cost of modifying it to suit your local needs can vary greatly. For example, WordML, generated by Word 2003, is complex and not very hierarchical, so customising it is difficult. Other tools have built-in support for DocBook or XHTML.
- **Software cost:** Most Word to XML converters are in the €100 - €500 price range, which is not very significant for single user licenses, but this can add up to a lot of money if a 100-user license is required.

## Case study: Portsmouth University

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



**Figure 4 Screenshot of Portsmouth University online course**

Portsmouth University offer a number of Masters degree courses by web-based distance learning. To minimise the cost of creating and maintaining course content, Pearson Education, who provided editorial and marketing services, choose to adopt the approach of authoring in Word, with automated conversion to final publishable HTML pages. They choose the YAWC Pro Word to XML converter developed by my company, XML Workshop Ltd., as the conversion tool, but in reality a number of other converters would also have been suitable. The actual Word to XML conversion is probably the most trivial part of the process, since off-the-shelf tools can be used. The more complex work involved the development of a supportive authoring and editing environment on top of Word, automatic generation of context-sensitive navigation bars, and conversion of interaction template tables into HTML, JavaScript and Flash-based online interactions.

A sophisticated Word template including custom menus, macros, toolbars and shortcuts, was used to provide additional authoring support on top of the basic Word environment.

Each course is created according to a Unit-Lesson-Topic structure, within a directory structure, with a standard file-naming convention for each file. For example, the Market Research Unit has the prefix 'mr', and all files in this unit have this prefix. Each Lesson is contained in its own numbered directory (e.g. Lesson 2 is mr02). Each directory contains all the topics for a lesson, with one topic per Word file.

Each Word document containing a learning object or topic is initially converted into DocBook XML format, and subsequently into HTML, using an XSLT stylesheet. We chose DocBook because it is a well-used and understood DTD, and many applications have been developed for it, such as typesetting

applications. Multiple web pages are created automatically, by breaking the original Word document at each top-level heading. Within these pages, we added a set of simple numbered hyperlinks to enable easy online navigation.

To create the left-hand navigation, we developed some VBA macros that recorded the titles and filenames of all the files in a Unit, and saved them in a single XML Table of Contents file, also using DocBook. When converting a particular topic into HTML, this file is read in, and converted to HTML, with the appropriate Lesson and Topic highlighted.

The most complex area of work was the generation of complete HTML and Flash-based interactions from Word tables. The tables are initially converted into DocBook `qandaset` elements and then into either a HTML form or a text file that can be loaded as parameters into a pre-defined Flash file for drag-and-drop style interactions.

## Conclusion

This paper describes a process that allows you to create and maintain accessible online learning materials simply and efficiently using Microsoft Word and XML. Adopting this process would allow you to dramatically improve the efficiency of the authoring process.

The technical cost of developing such a system is very low for the majority of material, but would be significant if you want to support the creation and maintenance of sophisticated interactions in Word.

The main organisational challenge lies in changing authoring habits and tools, and encouraging authors to focus on content, not presentation.

## Bibliography

- Joe Clark, 2002. *Building Accessible Websites*, New Riders Publishing, Indianapolis, USA. ISBN 0-7357-1150-X
- Jim Thatcher *et al.* 2002. *Constructing Accessible Web Sites*, glasshaus, Birmingham, UK. ISBN 1-904151-00-0
- Mark Pilgrim, *Dive into Accessibility*, <http://www.diveintoaccessibility.org/>
- Eoin Campbell, *Maintaining accessible websites with Microsoft Word and XML*, XML Europe 2003 (cf. <http://www.xmlw.ie/events.htm>)
- W3C, *Web Content Accessibility Guidelines 1.0*, <http://www.w3.org/TR/WAI-WEBCONTENT>