# Accessible File Formats in School Environments

## Introduction

As university faculty and elementary and secondary school teachers become more comfortable using technology in their classrooms, they are increasingly providing materials to students in electronic formats such as Portable Document Format (PDF), Hypertext Markup Language (HTML-the formatting code used to create web pages), word processing formats such as Word or Word Perfect, presentation formats such as PowerPoint, spreadsheet formats such as Excel, and graphical formats such as BMP, TIFF, and JPG. The most commonly used formats in schools tend to be PDF, HTML, Word and PowerPoint. For example, a professor might create a website or use a courseware management system such as Blackboard or WebCT to post materials so that students can access them when they are off-campus. Some of these materials might be posted in HTML. The professor might also upload documents such as articles and course notes for students to read. These documents might be in PDF or Word format. Finally, the professor might post the presentation overheads used during lectures. These files would likely be in PowerPoint format.

All of these formats can be created in ways that make them accessible to individuals with disabilities who use assistive technologies such as screen readers or text-to-speech software. However, without careful forethought, these materials can also be created in ways that block access for users with print disabilities (for example, blindness or low vision, learning disabilities, or motor control issues that prevent an individual from holding a book). In such circumstances, information technologies that could provide significant benefit to individuals with disabilities instead become an impediment.

## Who is Responsible for Ensuring that File Formats are Accessible?

There are at least five groups who have responsibility for ensuring that file formats are accessible to individuals with disabilities.

First, **content developers** (for example, the teacher, professor, web designer) must create files that are accessible.

Second, the **developers of rendering tools** (for example, the plug-ins that allow Internet Explorer to show Word or PowerPoint files) must create tools that don't modify the original document in ways that make it inaccessible.

Third, the **developers of assistive technologies** (for example, the screen readers and document reading software) must create their tools in ways that take advantage of accessibility information when it is available (for example, alternative text in word processing documents).

For more information on screen readers' support for file formats see the AccessIT Knowledge Base article: *How well do screen readers support web accessibility guidelines?*
http://www.washington.edu/accessit/articles?166

Fourth, the **developers of the file format** (for example, W3C which wrote the HTML specification, Adobe which wrote the PDF specification, and Microsoft which developed the Word file format) must ensure that the file format supports accessibility in the first place. W3C has done so with HTML. The others have also done so, but to a lesser degree.

Finally, **manufacturers of tools that create files in these formats** (e.g., web-authoring tools like Dreamweaver and FrontPage and PDF-authoring tools like Qwark) must ensure that if a file format supports

accessibility features, its tools support the creation of accessible file formats. Most PDF and HTML tools currently fall short of full support.

For more information on the accessibility of web authoring tools see the AccessIT Knowledge Base article: *Can I make accessible web pages using web authoring tools such as FrontPage and Dreamweaver?* http://www.washington.edu/accessit/articles?120

All five of these areas are important, but this brochure focuses primarily on those areas where content developers have the most control and responsibility.

## The Role of Content Developers

To ensure that a file format is accessible, content developers must do two things. First, they must ensure that text is available for use by assistive technologies. Second, they must ensure that the text available to assistive technologies has structural integrity.

A file format is inaccessible when it does not provide textual information that can be used by assistive technology devices or software. For example, when text is saved in formats such as JPEG or GIF (both formats for saving images), the text is inherently inaccessible for people who are blind or who have dyslexia because the format saves only a visual representation of the text. There is no information for screen readers or text-to-speech software to read. The same is true of some PDF files. When PDF files are created by scanning a page from a book, but not converted to text by using optical character recognition software (OCR), the resulting files are essentially pictures of the book. They provide no additional information to assistive technologies. A primary requirement of accessible file formats is that text must be available.

File formats are also inaccessible when textual information is available to assistive technologies, but not in a way that allows the user to make sense of the information. This happens when the **structural integrity** of the document is not clearly defined.

To understand this problem, consider a complex Word document that has columns, embedded images, footnotes, headers, and sidebars. Individuals who can see understand the purpose and function of each of these items based on their placement on the page; that is, through the visual layout or design of the page. They can see that the titles are all in bold, 14-point font, and that the footnotes are all in italics, 9- point font at the bottom of the page. The layout and visual design of the document gives cues about the structural logic of the content. For readers who cannot see, these cues are unavailable, so the structural logic of the content must be presented in another way. Without information about the structure of a document, screen readers tend to read linearly, moving from left to right. Figure 1 shows a complex document with the sections numbered to indicate the order in which they would be read if the document were created without structural integrity.

## How do File Formats Communicate Structural Integrity?

To create a file with structural integrity, users must use tags or styles that communicate the structure of a document to assistive technologies. For example, in Word, a content developer would need to carefully use styles to communicate structural integrity. Styles are a collection of formatting instructions used in word processing programs. For example, a content developer could set a style for Heading 1 to be bold, 14-point font, and Arial. Styles allow the developer to label features such as headings and footnotes and ensure that they are consistently formatted throughout a document. Figure 2 shows a document with three levels of heading. Each heading is formatted differently because a different Word style has been applied.

In HTML, a developer would use tags to denote structural integrity. For example, the tag <h1> denotes text as being Heading 1. Figure 3 shows an example of HTML coding; next to it is an example of how it would be rendered by a browser.

It is important to note that in both Word and HTML, a developer could achieve the same visual effect without structural integrity by simply changing the font style and size. Doing so would result in a document that has

visual structure, but lacks the underlying structural integrity needed by screen readers. For example, in Figure 2 a content developer could have used the bold and italics commands to create the visual appearance of Heading 2.

# Accessibility of Common File Formats

## HTML

File formats differ in their support of structural integrity. HTML is probably the format of choice when it comes to accessibility. Although it is possible for authors to create documents with little or no structure, HTML is a highly structured language and there is growing support for accessible markup.

## Microsoft Word

Microsoft Word allows developers to create structure through the use of styles, although authors seldom use them. Word is also problematic in the way it communicates to assistive technologies. Word has no HTML-like accessible table structure or forms structure. Even when graphics have alternate text associated with them, the text is not available to most screen readers when viewed in the Word application.

For more information on the accessibility of Word see the AccessIT Knowledge Base article: *How accessible are Microsoft Word documents?*
http://www.washington.edu/accessit/articles?266

## An important note about native applications versus viewers

From an accessibility standpoint, there is a big difference between opening a document in a native application versus opening it in a viewer. For example, you can open Word documents and PDF documents in a web browser such as Internet Explorer. When you do so, you are viewing the document using a viewer built into the browser, not in the native application (that is, not in Word or Adobe Acrobat). The viewer in the browser doesn't interpret the structure in same way as the native application and loses the structural integrity of the document. Screen reader users generally must open the documents in their native application to be able to interpret them. Therefore, content developers should always try to create documents in formats that they know their users will have on their desktop.

One of the reasons that HTML is a preferred format from an accessibility perspective is that the native application is the web browser (for example, Internet Explorer, Netscape or Mozilla), so there is no need to open a different application to view HTML files.

## PDF

There are three types of PDF files: unstructured (image), structured (embedded fonts), and tagged. Only tagged PDFs are optimized for accessibility. Tagged PDFs have an HTML like structure and support alternate text. They are easily created from Word documents if the Word document is itself correctly styled.

For more information on the accessibility of PDF, see the AccessIT Knowledge Base article: *Is PDF Accessible?*
http://www.washington.edu/accessit/articles?2

## Microsoft PowerPoint

PowerPoint is another popular file format, but it poses different accessibility problems and solutions. Like other file formats, PowerPoint files are reasonably accessible when viewed in the native application. Accessibility can be a problem when PowerPoint files are exported to the web.

There are many ways to export PowerPoint files to the web. One simple way is to post the original PowerPoint file. This has advantages and disadvantages. The advantage is that the user who has the native PowerPoint application on his or her computer can simply download the file and view it in the application. The disadvantage is that if the user does not have the PowerPoint application, it must be viewed using the PowerPoint plugin, a browser-based viewer. Unfortunately, the viewer does not render the files in a way that is usable by screen reader users.

Another common approach is to use the PowerPoint's "Save As Web Page" command, but this results in the creation of HTML files that are inaccessible in some browsers and not easily interpreted by screen reader users in other browsers.

A better approach is to purchase the Illinois Accessible Web Publishing Wizard for Microsoft Office (http://cita.disability.uiuc.edu/software/office/). This wizard assists the user in easily creating a variety of different versions of the PowerPoint presentation, including Text Only, Text Mostly, Graphic, Outline, and Handout versions. It also supports adding alternative text to images. Examples of these versions can be viewed on their website (http://cita.rehab.uiuc.edu/software/office/example/).

There are other ways to export PowerPoint accessibly to the web without using the wizard. For example, a content developer could copy the outline from PowerPoint, paste it into web authoring software and add HTML structure (e.g., markup bulleted lists and add <h1> for slide titles).

For more information on the accessibility of PowerPoint see the AccessIT Knowledge Base article: *How do I make my online PowerPoint presentation accessible?*
http://www.washington.edu/accessit/articles?28

# Next Generation Accessible Formats

New standards, specifications and guidelines have been developed in recent years to address accessibility of books and instructional materials. The development of these specifications is important because they ensure that content will eventually be interoperable across both service organizations and the developers of playback systems. The hope is these standards will be broadly adopted by publishers so that individuals with print disabilities will have easy access to electronic materials.

## DAISY

DAISY (Digital Accessible Information SYstem, www.daisy.org) is a technical standard for producing digital multimedia documents that are easily navigated and accessible. These documents are often referred to as Digital Talking Books (DTBs). DAISY was developed to increase access to textual material by people with print disabilities. DAISY 3 is a formal ANSI/NISO specification for digital talking books. Specifically, it is an XML language for adding structure to books (both textonly and text-with-synchronized audio). A growing number of production and playback tools can read DAISY files.

For more information on the accessibility of XML see the AccessIT Knowledge Base article: *Is XML Accessible?*
http://www.washington.edu/accessit/articles?26

## National Instructional Materials Accessibility Standard (NIMAS)

Whereas DAISY 3 focuses on accessible books of all types, the National Instructional Materials Accessibility Standard (NIMAS) addresses accessible instructional materials such as textbooks. NIMAS (http://nimas.cast.org/) is an extension of DAISY 3. It is endorsed by the U.S. Department of Education and will be the national standard adopted by the Secretary of Education, as specified in the 2004 reauthorization of the Individuals with Disabilities Education Act (IDEA) (http://nimas.cast.org/about/idea2004/index.html).

NIMAS addresses the difficulty that students with print disabilities have in acquiring accessible instructional materials. A high school student who is blind, for example, might wait several weeks for an accessible version of a textbook chapter to be produced in electronic form or Braille. This puts the student at a significant disadvantage in comparison to other students in the same class. NIMAS provides textbook developers with a standard approach to creating accessible electronic files.

Some States have adopted NIMAS. Visit the NIMAS Technical Assistance Center's website for an overview of legislation related to accessible instructional materials (http://nimas.cast.org/about/resources/legislation.html). Of particular interest, NIMAS has conducted a survey of state and territory laws pertaining to the provision of accessible materials for K-12 students with print disabilities (http://nimas.cast.org/about/resources/statessurvey.html).

For more information on state accessible textbook laws see the AccessIT Knowledge Base article: *Which states have accessible textbook laws and what do they say about file formats?* http://www.washington.edu/accessit/articles?243

| File Format | Strengths | Weaknesses |
| --- | --- | --- |
| HTML | • Highly structured language<br>• Structure facilitates navigation for screen reader users<br>• Growing AT support for accessible markup<br>• Native application is the browser, so files render correctly | • Possible for authors to create documents with little or no structure<br>• Not everyone knows how to create web pages using HTML or has access to a graphical web development tool |
| Word | • Commonly used and familiar<br>• Using styles allows a content developer to create documents with structural integrity<br>• Converts well to other formats when styles are used | • Easy to create the same visual effect using formatting (e.g., bold, italic) rather than styles<br>• Few people use styles to structure their documents<br>• Most screen readers don't effectively support alternative text for images within the native application<br>• When Word files are viewed using a web browser, the viewer will render the file poorly and loses the structural integrity of the document. |
| PDF | • Commonly used<br>• Viewers are ubiquitous<br>• Tagged PDF are accessible to many screen readers<br>• Supports alternative text for images | • Easy to create non-accessible PDF files<br>• Difficult to modify PDF files once they are created<br>• Viewer in browser will render |

| | | incorrectly, must open in native application |
|---|---|---|
| PowerPoint | • Commonly used and familiar<br>• Files are accessible within native application<br>• A Wizard is available for converting files accessibly to the web | • Doesn't convert well to the web when using typical approaches<br>• Viewer in browser will render incorrectly, must open in native application |
| DAISY 3 and NIMAS | • Highly structured XML files are very accessible<br>• Standards are becoming embedded in law | |