

## Blaise IS and Accessibility

*Jim O'Reilly (Westat, USA)*

### 1. Introduction

An important benefit of information technology is that persons with disabilities can participate in many activities and functions more fully than in the past. As the centrality of IT and particularly the Internet to communications has grown in the last two decades, the value and necessity of accessibility for the disabled has increased as well.

Many governments have policies mandating or promoting IT and Web accessibility. The W3C lists<sup>3</sup> government policies for the EU and sixteen countries<sup>4</sup> as well as those of Australian and U.S. states and Canadian provinces.

This presentation will focus on accessibility and Blaise in terms of the U.S. Section 508 standards for Accessibility for Electronic and Information Technology and People with Disabilities. While Section 508 is a U.S. regulation, its goals, standards and issues are quite general and applicable to Blaise users in other countries.

We will discuss the 508 requirements and process, review our experience with accessibility and the Blaise for Windows platform, and then focus on Blaise IS and Web accessibility.

### 2. Background

Section 508 is part of a Federal law requiring executive-branch agencies to provide comparable access to electronic and information technology (EIT) for its staff and the public with disabilities. The law creates an Access Board and a Federal Acquisition Regulation (FAR) to determine how to meet the requirement.

Most of the specifications for software concern usability for people with vision impairments (alternative keyboard navigation, animated displays, color and contrast settings, flash rate, and electronic forms, among others).

Section 508 Web-based technology and information are based on the Web Accessibility Initiative of the World Wide Web Consortium. These ensure access for people with vision impairments who rely on various assistive products to access computer-based information, such as screen readers, which translate what's on a computer screen into automated audible output, and refreshable Braille displays. Verbal tags or identification of graphics and format devices, like frames, are necessary so that these devices can "read" them for the user in a sensible way.

The standards do not prohibit the use of web site graphics or animation but require that such information is also available in an accessible format. The specification also addresses the usability of multimedia presentations, image maps, style sheets, scripting languages, applets and plug-ins, and electronic forms<sup>5</sup>.

---

<sup>3</sup> [www.w3.org/WAI/Policy](http://www.w3.org/WAI/Policy)

<sup>4</sup> Australia, Canada, Denmark, Finland, France, Germany, Hong Kong, India, Ireland, Italy, Japan, New Zealand, Portugal, Spain, United Kingdom, United States of America

<sup>5</sup> <http://www.section508.gov/index.cfm?FuseAction=Content&ID=11>

### 3. Blaise for Windows and Screen Readers

Blaise and Westat first focused on 508 and accessibility in terms of the Blaise for Windows platform. The approach was to rely on Microsoft Windows® support of accessibility and universal design and on the Blaise system's rigorous implementation of Windows standards. Westat also published a statement on Blaise and accessibility, [www.westat.com/blaise/508info/blaise508stmt.htm](http://www.westat.com/blaise/508info/blaise508stmt.htm)

The Blaise team and Westat also committed to addressing accessibility issues that are reported by customers and users. In 2003 a customer reported that a blind application software tester could not make Blaise applications run in the Windows environment with the JAWS screen reader.

We confirmed the problem and investigated possible solutions. We quickly learned that custom-developed scripts for JAWS are common. The scripts specify for JAWS how specialized elements of an application's user interface work and how they should be treated by the screen reader. Once a script is developed and installed, JAWS will automatically use the script when processing specific types of files based on the file extension.<sup>6</sup>

Westat contracted with a consultant experienced in JAWS scripting for development of a JAWS script for the Blaise Data Entry Program (DEP) user interface. The consultant developed and delivered the script expeditiously, saying only a few elements of the DEP UI were problematic for JAWS. We tested DEP applications in JAWS and then passed the script along to the original blind software tester. She reported that it made DEP applications fully functional with JAWS.

Blaise for Windows applications were demonstrated running in JAWS with the script at the last IBUC conference in Ottawa, September 2004. The script is available from [blaise@westat.com](mailto:blaise@westat.com).

### 4. Blaise IS (Internet Services)

Web surveys present significant challenges and opportunities for accessibility. Key elements are:

- The Browser environment is diverse, changing and controlled by the user.
- Survey users are often the general public not just agency staff or employees. User experience, training, and support can't be assumed.
- The Web offers rich presentation features and user-interface elements that must be communicated properly by the screen reader.

The dynamic, diverse nature of web survey content also means that accessibility testing and evaluation must be pro-active and continuing rather than waiting for problems to be reported. Survey applications produced by the same system and structure may look and function quite differently because of the flexibility of the Web presentation. As a result, testing needs to take place at the application level, as well as the general system level.

---

<sup>6</sup> We also learned that a lively community of consultants experienced in JAWS scripting can be found through [www.blindprogramming.com](http://www.blindprogramming.com) and other sites.

## 5. IS architecture – a higher level approach

The architecture of Blaise IS provide a major advantage in terms of the ability to implement accessibility features in Web surveys. IS is built on the technologies of ASP, XML and XSL. Little or no direct HTML coding is done in authoring an IS survey data model. The developer works in the Blaise Control Centre and builds the data model in same way as Windows CAPI or CATI—coding the familiar fields, rules, blocks, edits, etc.

Web presentation and look-and-feel is implemented with the settings in the mode library and LAYOUT statements in the data model. Other web survey characteristics are set outside of the data model in the specification file—the start and end ASP pages, the prepared data model, the XSL interview style sheet to generate HTML pages, and others elements.

Once the IS survey package is installed and active on the server, the interviewing process proceeds:

- The IS server receives requests and responses submitted by the web survey user.
- Data model rules are run.
- The next page's information is generated in a XML document.
- The document is passed to a process that applies the specified IS interview XSL style sheet and generates the HTML page.
- The HTML page is sent to the user.

So the XSL interview style sheet, as the generator of the HTML document, is the key to survey look-and-feel. It is the single location for implementing accessibility enhancements. Addressing web survey accessibility issues generally mean adding appropriate HTML tags or attributes to enable the screen reader to interpret the page properly. With Blaise IS this can largely be accomplished by adding commands to the XSL interview style sheet to generate these elements and their values. Once defined, the XSL interview style sheet can be used for future surveys with no further effort.

It's important to understand how large and sophisticated the primary IS interview style sheet is. There are some 9,000 lines of code, complex structures and many intricate relationships. Although it is a text file in a standard language that others could edit on their own, in practice only specialists on the Blaise development team should make changes. This is because they know how, because they will maintain the document, and because the evolving document will be distributed with each update of the system.

It is hard to overstate the importance for accessibility of the Blaise IS architecture and the interview style sheet. It provides an extensible capability well suited for flexibly improving and evolving the accessibility features of the system.

## 6. Section 508 Standards

Section 508 standards are comprehensive and general, as illustrated by Figure 1 for Web-based information and application.

**Figure 1: Section 508 Standard 1194.22 for Web-based intranet and Internet information and applications.**

- (a) Text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content).
- (b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.
- (c) All web pages information conveyed with color also available without color, e.g. from context or markup.
- (d) Documents organized so they are readable without requiring an associated style sheet.
- (e) Redundant text links provided for each active region of a server-side image map.
- (f) Client-side image maps provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.
- (g) Row and column headers identified for data tables.
- (h) Markup used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.
- (i) Frames titled with text that facilitates frame identification and navigation.
- (j) Pages designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.
- (k) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.
- (l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.
- (m) Web page requiring that an applet, plug-in or other application on the client system to interpret page content, must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).
- (n) Electronic forms designed to be completed on-line shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.
- (o) A method shall be provided that permits users to skip repetitive navigation links.
- (p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

## 7. Accessibility testing of Blaise IS Surveys

In the first stage of our work on accessibility and Blaise IS, we focused on automated software to analyze web pages and report whether the Section 508 standards are being met. There are a number of free services on line and commercial systems that will check how well a web page meets the 508 web standards. W3C has a list of such tools<sup>7</sup>.

---

<sup>7</sup> <http://www.w3.org/WAI/ER/tools/>

Because IS surveys may have hundreds of pages, we looked for an application able to test many pages in a batch. We selected AccVerify ([www.hisoftware.com](http://www.hisoftware.com)). Next, to capture the Blaise IS pages for a survey session we adapted the standard IS ASP page handler (biPagHan.asp), adding code to save every page to a file.

Our first test was of a sizeable web survey that had been completed. In the test session 182 web pages were saved and then processed through AccVerify. The testing reports identified that every page with an enumerated question “failed” because the radio button graphical display did not include an “alt=” string.

**Some local television news programs include special segments of their newscasts that focus on health issues. In the past 12 months, have you watched health segments on the local news?**

Yes

No

---

Don't know

Rather not answer

We discussed the issue with the Blaise team and agreed on the text to put in the alt= string. The XSL interview style sheet was revised to add this. We recorded the survey again and passed the recorded pages to AccVerify. No problems were reported.

We repeated the cycle a few more times—testing web surveys, processing pages with AccVerify, finding failures of different types, conferring with the Blaise team, receiving a revised interview XSL styles and retesting to confirm the change worked.

This process clearly illustrated the flexibility and power of the Blaise IS XML/XSL architecture in that significant enrichment of the system for accessibility could be implemented simply and quickly. At the same time we learned of the limitations of the automated testing of accessibility and the importance of testing with a screen reader.

## 8. Screen Readers

Screen readers are the essential assistive technology for blind users. Screen readers applied to a browser read the html document’s text, interpret the tags and other elements, and convert the “readable” material into synthesized speech. From testing pages on the JAWS system and discussions with experienced users it’s clear that

- Screen readers handle properly some things AccVerify and similar 508 standards checkers fail, e.g. radio buttons.
- Conversely, screen readers can be stumped by survey elements that meet the specific 508 standards.
- The true test for accessibility for blind users is whether the screen reader is able to make the page sensible for the user.

Another significant issue is that using a screen reader can be difficult for the inexperienced. Screen reader users rely heavily on key-combinations to navigate about the page. JAWS, the mostly widely used product in North America, has

about 66. So a sighted person with limited JAWS skills may struggle to determine if a problematic read of a page is the fault of the code or the JAWS user.

## 9. Screen Reader Testing of Blaise IS

We conducted two tests with JAWS of Blaise IS surveys included among the samples provided with the system. These were Expenses and NSSE2003. One tester was a staff member with normal vision and a few hours experience with JAWS. The second was an accessibility consultant who is blind and experienced with JAWS and other screen readers.

**Figure 2. Tests of IS surveys with JAWS screen reader**

*Test 1: sighted Blaise developer*

- Reads header but not footer (Prev, Next buttons)
- Items with preface text—not reading preface
- At times not reading enumeration sets past first empty selection
- Tables—column labels not read
- Items with icons to left (Help & Comment buttons) are not read properly

*Test 2: (blind consultant)*

- Overall 70-80% of full accessibility
- Need explicit titling of columns with <TH> tag
- Form elements need a <Label for="idname">
- Recommends alt-key combinations for key navigational elements, <Prev>, <Next> etc.

Of the items, only one was identified by both tests—labels for table columns. For the rest of the items mentioned by the sighted JAWS user, the blind user didn't report a problem. This effort clarified two issues: first, inexperienced JAWS users have limited value as testers and, second, the ad hoc approach of finding problems here and there and fixing them is insufficient. The number of issues that need to be considered is larger than initially believed.

## 10. A Broader Approach to Web Survey Accessibility

Interpretation and implementation of the 508 standards has been evolving as Federal agencies and others analyze the issues, develop interpretations, specify requirements and test products. The most detailed and well organized that we have found is from the U.S. Social Security Administration (SSA). As an agency with the elderly and disabled as major constituencies, SSA has taken the lead in developing processes and materials to communicate to government officials, software developers and vendors how to apply Section 508.

SSA's Accessible Solutions and Usability Branch has produced a significant set of materials related to accessibility. In the "Guide to Applying Section 508 Standards", the standards are expanded upon with SSA's interpretation of the standard and the specification of one or more requirements to meet the standard. For each requirement the assistive technology to be supported, recommended evaluation criteria and the testing method to judge whether the requirement is met are all provided. Figure 3 presents the SSA information on the first of fifteen elements of the Web-based intranet and Internet information and applications standard Sect 1194.22.a.

Figure 3: SSA Elaboration of Section 508 Section 1192.22.a Standard<sup>8</sup>

<b>Standard 1194.22.a</b>	<b>A text equivalent for every non-text element shall be provided (e.g. via “alt”, “longdesc”, or in element content).</b>	
<b>SSA Interpretation</b>	AT users shall be able to access all meaningful non-text elements.	
<b>Requirement (a.1)</b>	<b>Screen readers shall be able to speak all alt text or equivalent elements.</b>	
	<i>Supported By</i>	JAWS
	<i>Recommended Evaluation Criteria</i>	All alt text must be meaningful and speak when accessed.
	<i>Testing Method</i>	Tab through controls to ensure that they speak and are meaningful. Arrow through images to see that they speak and are meaningful.
<b>Requirement (a.2)</b>	<b>Keyboard and voice input users must be able to access all text equivalent alternatives for non-text elements.</b>	
	<i>Supported By</i>	JAWS
		MAGic
		Dragon
	<i>Recommended Evaluation Criteria</i>	All alt text or any equivalent must be exposed by the keyboard or voice input.
	<i>Testing Method</i>	Navigate by keyboard to see whether alt text or tool tips are exposed by using the keyboard or voice. ASB has developed a JavaScript capability to accomplish this.
<b>Requirement (a.3)</b>	<b>If textual links are not possible, then the screen name of the non-text element will be at the beginning of the alt-text tag, long description, title attribute, etc.</b>	
	<i>Supported By</i>	JAWS
		MAGic
		Dragon
	<i>Recommended Evaluation Criteria</i>	All text links must have alt text beginning with the screen text shown.
	<i>Testing Method</i>	Use AT to expose alt text using screen readers, keyboard, or voice input.

The SSA Guide to Applying Section 508 Standards covers all six standards. Two of the six standards relate to survey applications including Blaise IS:

- 1194.21 Software applications and operating systems, 11 sections (a-l) and 56 requirements;
- 1194.22 Web-based intranet and Internet information and applications, 15 sections (a-p) and 54 requirements.

<sup>8</sup> Section 508 Accessibility for Electronic and Information Technology for People with Disabilities: Guide to Applying Section 508 Standards: U.S. Social Security Administration (undated).

Of the 110 total requirements for the two standards, the following appear relevant to Blaise IS and accessibility with JAWS.

**Figure 4: Requirements most relevant to Blaise IS and use of screen readers.**

Section	Requirements
21a	<p>1, 2: Hotkeys must be assigned to commonly used control elements or notebook tabs to provide comparable mouse access. Tabbing which results in excessive keystrokes (more than 3) to activate an element is considered non-compliant. Hotkeys must be assigned to links used repeatedly on more than one screen or identical links used as templates for multiple applications.</p> <p>3, 4: Tab indices must be assigned to significant text information, directional cues, and error text information, or a comparable access method must be attained. Must be a logical tab order, generally left to right, top to bottom.</p> <p>6: Navigation to screen elements with the keyboard/voice must be available in a comparable manner to a mouse.</p> <p>8: Hotkeys must not conflict with the browser.</p> <p>10: Keyboard must be used to open list boxes, open and close tree structures, and navigate to the next logical item in proper order.</p> <p>11: Navigation to radio buttons using arrow keys, items in a combo box, list views, or check boxes must not automatically select the item and/or change focus.</p> <p>12: When focus changes visually to a new frame/pane or a logical place on the screen, a keystroke must be available to move focus if a well-defined visual focus does not occur.</p> <p>13: When embedded links are within sentences or paragraphs, the link must stand on its own to be meaningful <i>and</i> must have a title attribute that is meaningful within the context of the sentence.</p>
21c	<p>1: AT users shall be able to use the keyboard or voice commands to move to logical points of focus in order to attain focus on selected items.)</p>
21d	<p>1, 2, 3: Navigation to controls must result in appropriate speaking of labels, data and cues; a well-defined visual focus; and track with screen magnifiers.</p> <p>4: Interface elements must be exposed to voice recognition technology so users can access all controls.</p>
21f	<p>1: Screen readers should speak all significant text displayed</p>
11	<p>1: Screen readers must speak all controls, labels, directions, and cues in a logical order.</p> <p>2: Keyboard users must get focus to all controls, directions, and cues.</p>

	<p>3: Errors can be handled by the use of pop-ups, “red balls” or any other means. Errors can also be displayed after inputting a field or submitting a page. Access to errors must be keyboard accessible and comparable to a mouse user.</p> <p>5: The information within the pop-up should speak immediately through a screen reader.</p> <p>6: The pop-up should receive focus for screen magnification software.</p> <p>9: A list of errors must be identified at the top of the page and receive focus. Links to each error must be displayed at the top of the page.</p> <p>10: Each error on the page is associated with the element in error by some indicator that is identified by speech or text.</p> <p>11: The user must have the ability to navigate precisely to each identified error on the page and know precisely where each error is without having to navigate through an entire form.</p>
22a	1, 2, 3: Text equivalent for all non-text elements
22g	1: Row and column headers identified for data tables
22n	<p>1: Screen readers must speak all controls, labels, directions &amp; cues in a logical order.</p> <p>2: Keyboard users must get focus to all controls labels directions cues.</p> <p>3, 5, 6, 7: Errors can be handled by pop-ups “red balls” or other means. A list of errors shall be identified at the top of the page and receive focus. The user shall have the ability to navigate precisely to each identified error and know precisely where each error is without entire form navigation.</p>
22o	1, 2: Provide links/methods to skip over redundant links.

These 37 requirements group roughly in five areas:

- Speaking all controls, labels, directions, non-text elements, and in a logical order
- Hot keys and tabbing
- Navigation
- Tables
- Error handling

Some of these elements are currently in the system. But others may be challenging to implement. Fortunately, the XML, XSL architecture of the Blaise IS system provides a strong platform for extending the system to meet these challenges.

As well, the information available from the Social Security Administration provides valuable direction and support for the effort. Not only are the issues and requirements specified, but evaluation criteria and testing methods are clearly defined.

With this and the skill and dedication of the Blaise team, there is reason to be optimistic that Blaise IS’s current “70-80%” level of accessibility for the visually

impaired can be raised to the 9x% level. Both the Blaise team and Westat will be working on this effort in the coming period.

## **Session 2.4: Challenges at Statistics Netherlands**

### **The Dutch Annual Business Inquiry: Developing and Testing an Electronic Form**

*Ger Snijkers, Evrim Onat, Jo Tonglet, Rachel Vis & Robert 't Hart  
(Statistics Netherlands & Metaform, Amsterdam)*

### **CASI questionnaires with Blaise 4.7**

*Gerrit de Bolster  
(Statistics Netherlands)*

### **Basil, A New Tool for CASI in Blaise**

*Roger Linssen & Jo Tonglet  
(Statistics Netherlands)*

