

Developing Accessible Blaise Surveys

Karen Brenner, Richard Frey - Westat

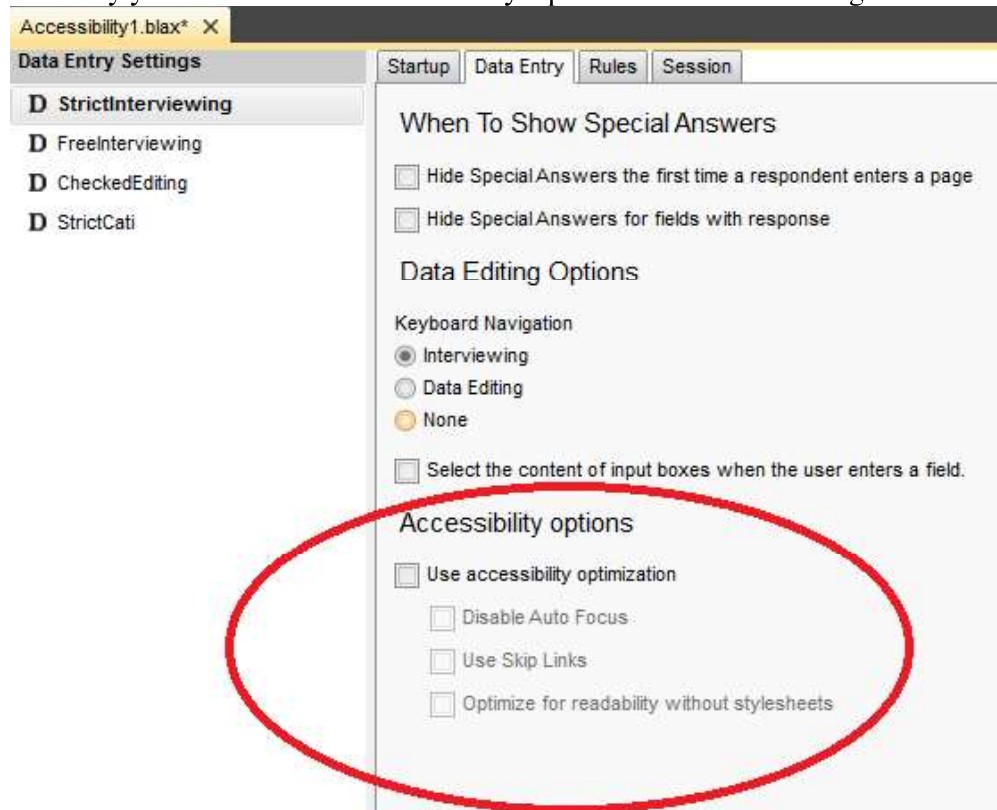
Accessibility and Universal Design

The ultimate assurance of accessibility, usability, and inclusivity across any system is Universal Design. Any survey requirements document can specify universal design to ensure accessibility for all users. Likewise, the Blaise programmer can develop the survey to meet these basic requirements. WCAG 2.0 are the international conformance requirements for accessibility, and the Section 508 Refresh (of the Rehabilitation Act) are the equivalent of those requirements in the United States.

Accessibility is coding your survey to the widest possible audience, including those with physical, cognitive, neurological, visual or auditory disabilities. Usability is the extent to which users can achieve specified goals effectively, efficiently, and with satisfaction. Universal Design is creating surveys that all people regardless of age, size, and ability can easily access, understand, and use to the greatest extent possible. All users benefit when a survey is accessible, usable, convenient, and enjoyable.

Techniques

Currently you need to turn on Accessibility Options in the Blaise Settings:



(Screen shot from v5.4.3.1675)

Activating the Accessibility options will turn on certain features within Blaise, and the developer adds the rest. To follow your requirements to meet the conformance requirements for Section 508/WCAG 2.0, follow all aspects of POUR: Perceivable, Operable, Understandable, and Robust. You can follow the techniques below to resolve some of the most common errors that we see in surveys:

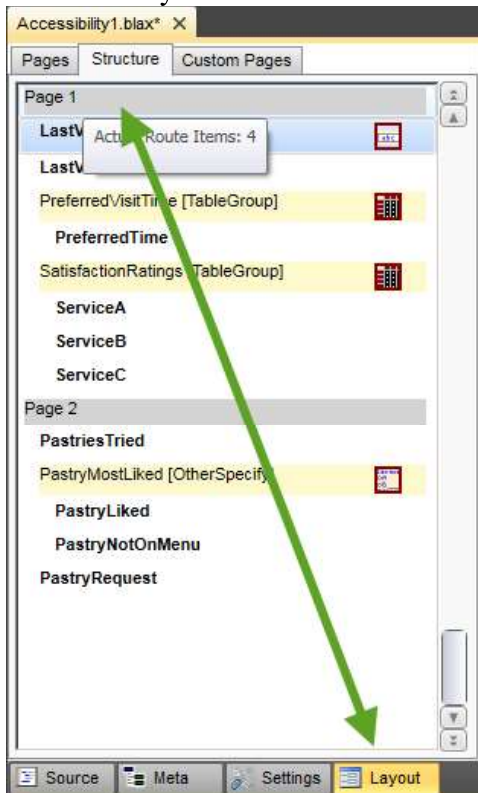
Color

The contrast of foreground text on a background must conform to minimum contrast requirements, and you cannot use color to convey meaning.

Added Colors

Set your design colors to pass contrast requirements.

1. Select the Layout view tab in the control center.



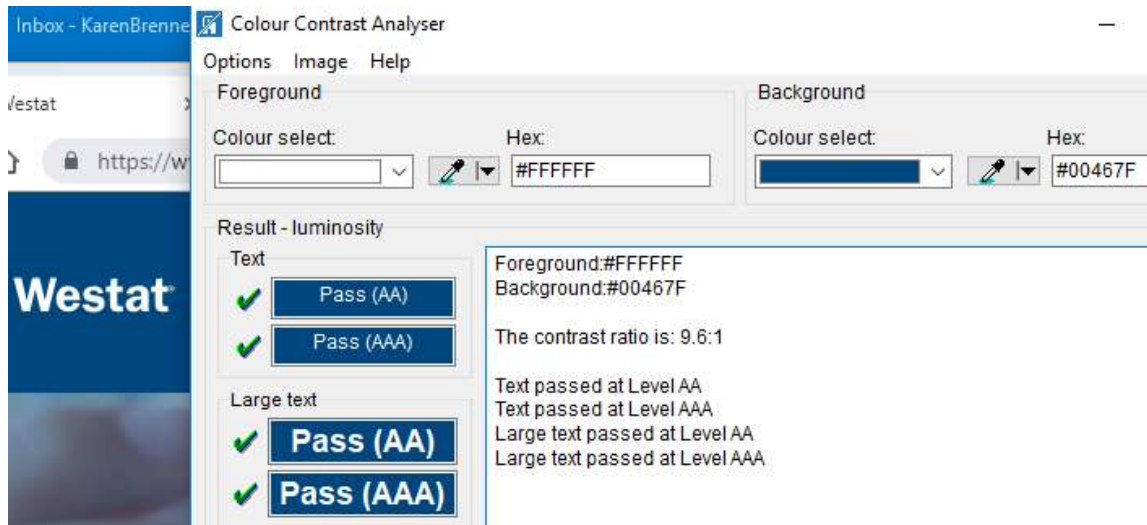
2. Find all color settings for the survey in the Resource Database. In the ribbon bar of the Layout view, most of the groups have a tiny little button in the bottom right corner. Click that to open the Resource Database Editor.

For example, in the Master Pages group, the red arrow points at the little “open the Resource Database Editor” button.



After selecting the Master Pages button, the Resource Database will open to the layout templates for the Master Pages. Each master page will have color properties such as the background color of the survey pages. The colors can be tested either by analyzing the hex value from the property's value or the display of the page in a browser. There are many color analyzer tools available to test your colors; one tool is the Color Contrast Analyzer v2.5, June 2017

<https://www.paciellogroup.com/resources/contrastanalyser/>



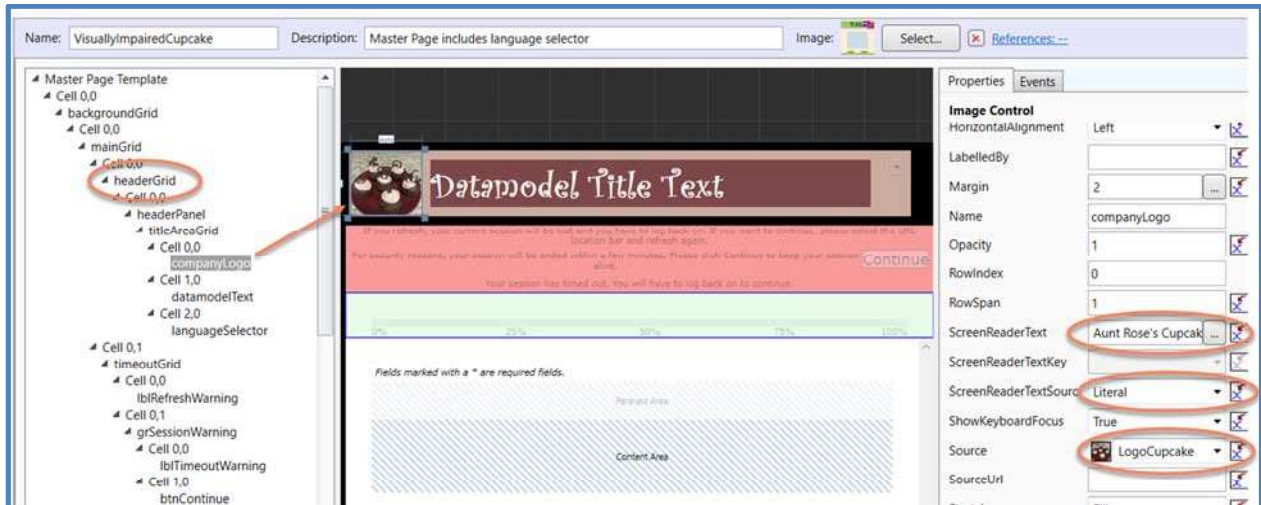
Default Colors

You may need to adjust, in the Resource database, the default colors for displayed text such as error messages. Navigate to Blaise Resource Database | Font Definitions | Controls | Error Text. On the Font Definition tab, change the Color property from #FFFF0011 (the default red) to “DarkRed” (one word - #8B0000).




Alt Attributes

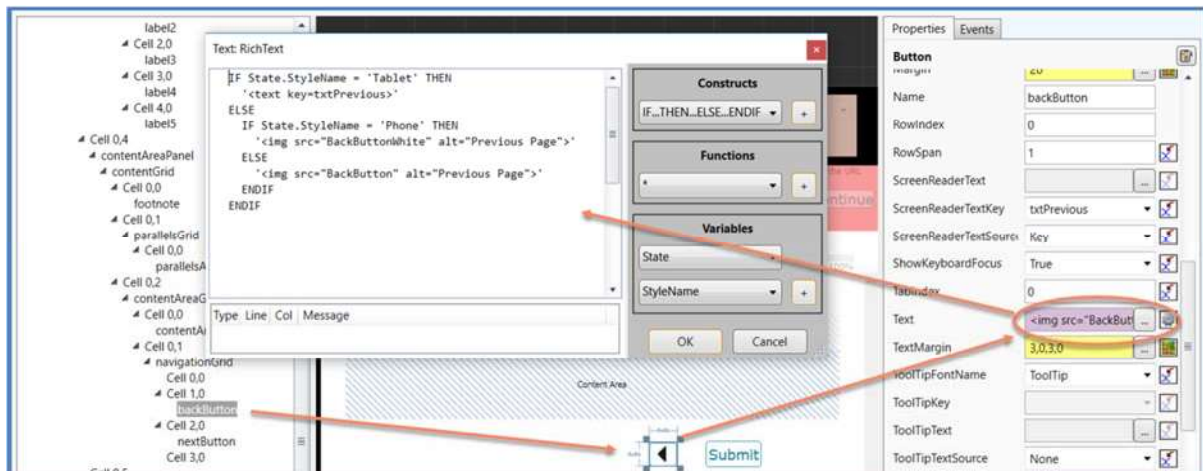
Images used in web surveys must have a description. This is accomplished by using the alt="" attribute. Accomplish this in Blaise by opening the Master Page in the survey’s Resource Data Base. To add a description for the company logo, for example, select the companyLogo Image Control in the headerGrid. Select the ScreenReaderTextSource dropdown property and pick Literal. The ScreenReaderText is now be enabled. Enter text, and save the resource database. In the example below, our alt attribute is alt="Aunt Rose’s Cupcake Company Logo".



Filling the alt attribute for other types of images, such as navigation buttons, may require a different method than the Company Logo. If you will only be using images for the navigation buttons then following the steps for the Company Logo will work. However, there may be a need to switch between layout styles, such as from a tablet style to a phone style, where the tablet might have a button using text and the phone using images. If this is the case, then you need an Expression to display the appropriate navigation button.

To add an Expression, first open the survey's Master Page in the Resource Database. Select the backButton Button Control in the navigationGrid. Select the Text property, right click on the  icon and select Expression. Now add an expression similar to the one below or build your own by using the Expression Editor to the left of resulting expression text.

The resulting expression will display the appropriate button as either predefined text or an image. The user needs to add the proper Alt text if the button is an image.



Each style shown in the expression will be associated with a Layout Set defined in the Layout Designer.

Headers

There are usually three header types in a survey: the page header, a section header and a sub-section header. Code header elements `<h1 . . . h6>` to the header types in the source. Code the header element for the page header in the Datamodel description text.

```
DATAMODEL FlightSurvey "<h1>Flight Survey ^PageTitle</h1>"
```

The code translates to the page header element as shown in this example from the Flight Survey:

```
<div class="text-container no-auto word-break">
  <div class="">
    <span>
      <h1>Flight Survey Page 1</h1>
    </span>
  </div>
</div>
```



Add header elements to the fieldname with a block type or a Groupname description for both section and sub-section headers.

FIELDS

```
Person "<h2>Personal Information</h2>" : BPerson
Flight "<h2>Flight Information</h2>" : BFlight
```

```
<div class="text-container no-auto word-break">
  <div class="">
    <span>
      <h2>Personal Information</h2>
    </span>
  </div>
</div>
```

Personal Information

?
What is your first name?

?
What is your last name?

The section and sub-section headers are considered a Group in the layout designer and would be assigned the Section group template and the Subsection template respectively.

Properly Set Properties

Some users might take a survey using the **assistive technology (AT)** of a screen reader. When a screen reader encounters an interactive element, for example a radio button, it needs to find all information related to that button. If not, the screen reader will repeat something like,

“Very satisfied radio button not check
 Satisfied radio button not checked
 Not satisfied radio button not checked
 Very satisfied radio button not check
 Satisfied radio button not checked
 Not satisfied radio button not checked
 Very satisfied radio button not check
 Satisfied radio button not checked
 Not satisfied radio button not checked”

In this example, the user cannot distinguish between the different choices as they arrow through a grid. Whether using Aria or Fieldset/Legend, it is important that the Name of the property include the question text, so the AT transmits all relative information to the user.

Note that this topic is one that exemplifies how there are different methods you can use when coding to conform to accessibility standards. You can always find the best techniques for accessibility conformance at <https://www.w3.org/TR/WCAG20-TECHS/>.

Blaise has provide three properties on many controls to assist with providing details about interactive elements.

You can identify sections of a page (landmarks) using the **AccessibilityRole** property. Landmarks help AT users orient themselves to a page and help them navigate easily to various sections of a page. Use the drop-down list to select a section of a page.

Use the **LabelFor** property to explicitly associate the label to a form control and improve accessibility.

The **LabelledBy** property of a control may contain a name of another control within the same template associated with this control. The mentioned control can contain, for example, an explanatory text.

The following enumeration question shows how the Labelfor and Labelledby set up the readable text needed by a screen reader in the Flight Survey:

Would you like to have an aisle seat next time?

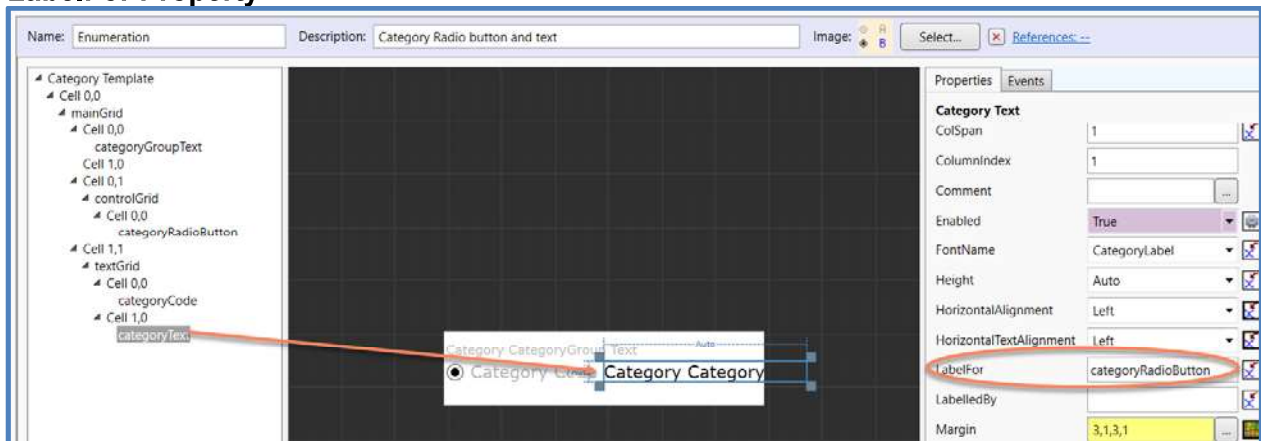
- * Yes, please I would like to have an aisle seat next time.
- No, thank you I do not want an aisle seat next time.

The following is the resulting aria-label with the question and category text:

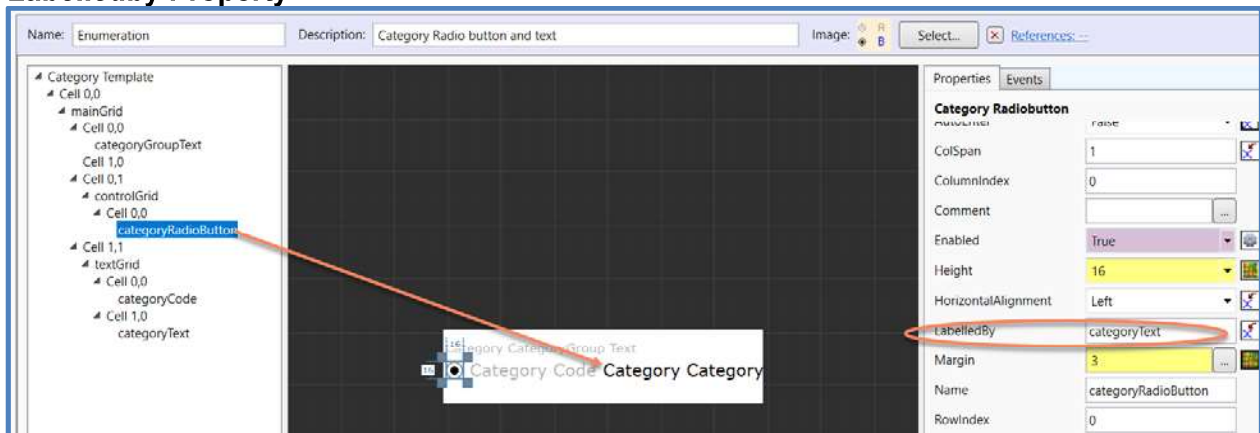
```
<input name="Flight.FirstClass" tabindex="0" class="Category  
RadioButtonComponent visibility-visible enabled focus-outlin  
e width-pixels height-pixels" id="aga_1ca_2gaba_1e" role="ra  
dio" aria-checked="false" aria-hidden="false" aria-invalid  
="false" aria-required="true" aria-labelledby="aga_1ca_2gaba  
_1h" aria-live="assertive" style="margin: 3px; width: 16px;  
height: 16px; visibility: visible;" aria-label="Would you li  
ke to have an aisle seat next time? Yes, please I would like  
to have an aisle seat next time." type="radio"  
data-fieldname="Flight.FirstClass" />
```

To achieve these results the Enumeration template must have the Labelfor and Labelledby properties set:

LabelFor Property



Labelledby Property

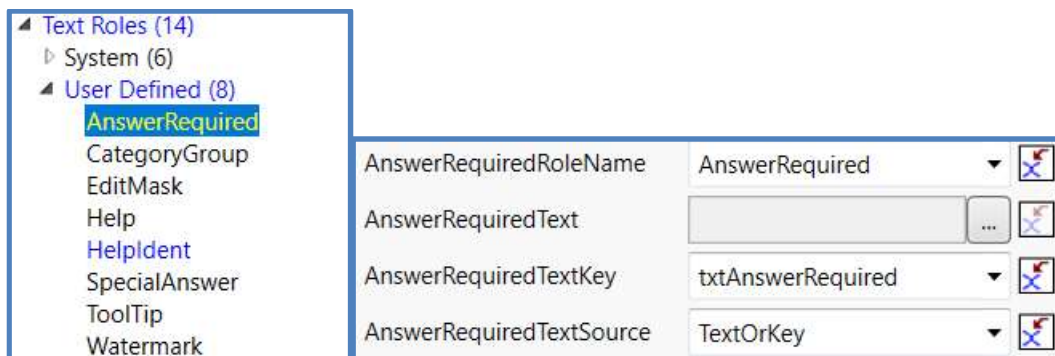


Required Field Error Messages

Another interactive element an AT user will encounter is the “Answer Required” error message. This standard message displays when a respondent does not enter a value in a required field. A sighted person would easily associate the displayed message “Answer Required” to the question that was not answered. A respondent listening to the survey with AT will need to understand which question was not answered if there are multiple questions on a page.

Blaise provides the ability to customize “Answer required” messages for each field that has a required attribute. Accomplish this using the AnswerRequired role text and a set of AnswerRequired properties of the ‘error text message’ found in many of the FieldPane templates.

The AnswerRequired role text is located in Role Texts|User Defined and comes with the default resource database. The AnswerRequired properties are found in the ‘Error Message Text’ of a field pane template.



To customize the text of error messages, the following properties are used:

- With the *AnswerRequiredRoleName* property, you can decide which control's Text Role you want to use as an error text.
- You can provide literal text that displays as soon as an answer is required with the *AnswerRequiredText* property.
- The *AnswerRequiredTextKey* property allows you to provide a user-defined translatable text.

- You decide whether there should be an answer required error text with the *AnswerRequiredTextSource* property. If so, you choose which bound element should be its source.

Next, add AnswerRequired to the Roles section in the datamodel.

```
ROLES = Help "The Help role provides a help button that the user can click to display question help text.",
        HelpIdent "HelpIdent helps to identify the help button for blind and partially sighted people.",
        Watermark "The watermark role provides a hint to the user as to what input elements are used for.",
        Tooltip "The Tooltip role provides a popup hint when the user hovers over a control.",
        AnswerRequired "The AnswerRequired role can be used to point out to the respondent that a question
                        has not yet been answered."
```

Then add the AnswerRequired role text to the questions with the NoEmpty attribute for which you want to clarify the AnswerRequired message.

```
LastName "what is your last name?"
  AnswerRequired "Your last name has not yet been entered. To continue, your last name must be entered."
  Help "Please enter your family name."
  <newline>Your privacy is guaranteed. Your name is only important to provide good service."
  HelpIdent "Help button for the question about your last name."
  : STRING[20]

Age "What is your age?"
  answerRequired "To continue, your age must be entered."
  Watermark "0..150" : 0..150

Town "Where do you live?"
  answerRequired "Your hometown is not yet known. Enter your hometown." : STRING[20]
```

The screen shot below shows an example of different Answer Required messages.

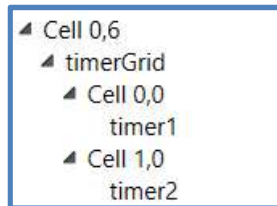
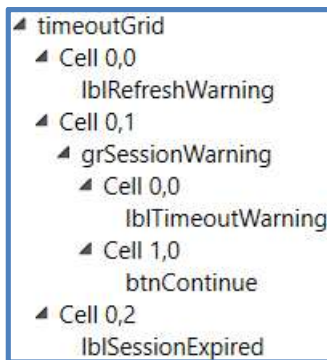
The screenshot displays three examples of 'Answer Required' error messages:

- What is your last name?**
Your last name has not yet been entered. To continue, your last name must be entered.
[Empty text input field]
- What is your age?**
To continue, your age must be entered.
[Input field containing "0..150"]
- Where do you live?**
Your hometown is not yet known. Enter your hometown.
[Empty text input field]

Setting notification of session expiration

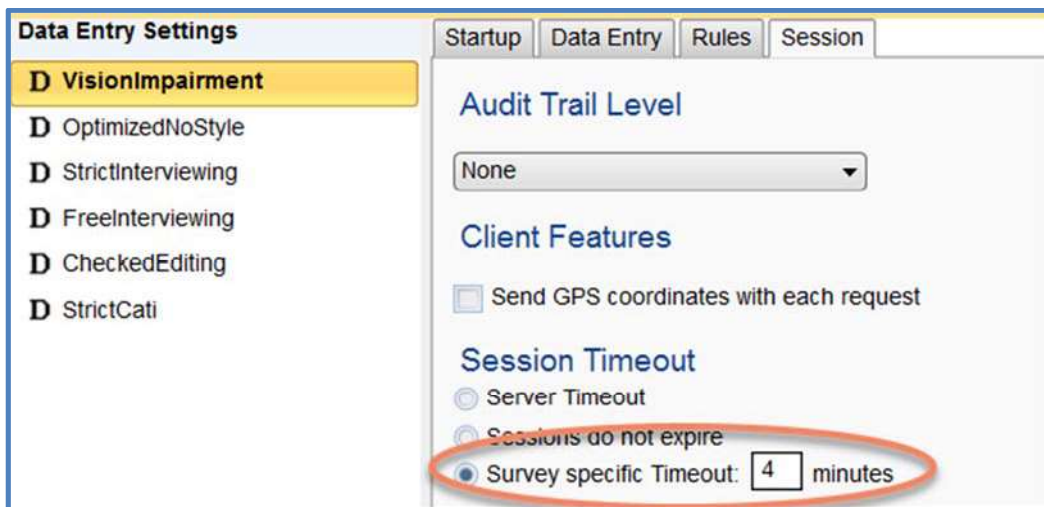
AT users need a warning when their session is about to expire. In some cases, they may need to re-authenticate to continue their session, and they want to avoid any loss of any data. If someone reads slowly or has trouble navigating, they may require additional time to complete an activity.

Blaise 5 provides an example of warning messages and the use of timers to warn users about the status of their current session. You can copy the session warning messages and timer structures from the VisuallyImpaired resource database and inserted into a survey's Master Page.

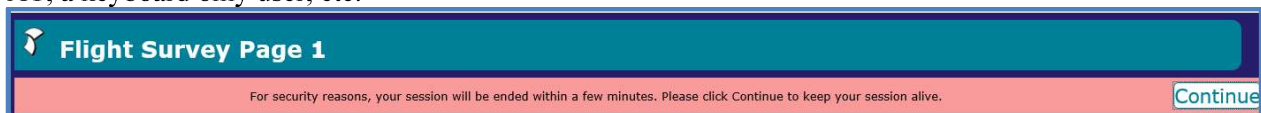


Copy the time Grid(Timeout grid) and the Grid(TimerGrid) from the VisuallyImpaired sample's blrd into your master page. To do this, add a row after the header in your master page. In the VisuallyImpaired master page, right click on Grid(TimerGrid), click copy, then go to your new row in your master page, right click and paste. Repeat the same step for the Grid(TimerGrid) but add the new row below your Navigation grid.

If the server session timeout is too long, the user can adjust the timeout in the Session tab of the Control Centre Settings.



Extend the Timeout depending on the length of your page to accommodate a slow reader, someone using AT, a keyboard only user, etc.



The warning messages should appear indicating the action the respondent can take, allow the user to take action using their keyboard, and provide enough time for all users to take action.

Summary

These are some examples of the many ways a developer can make surveys accessible. After addressing the accessibility issues described here, our survey passed Section 508/WCAG 2.0 testing. The tester used

a combination of manual testing and tools recommended by the Department of Homeland Security Trusted Tester certification training. The only outstanding issues were those related to readability without style sheets. Although Blaise relies heavily on style sheets, the Access Board proposed 1194.22(d) related to style sheets be removed from the conformance criteria since assistive technologies now support CSS very well. Blaise has made many improvements in v5.4 that allow designers and developers to create accessible surveys.