Making Documentation Accessible to Users With Disabilities

Gail B. Chappell

Good documentation takes into account the needs of people with disabilities. Such documentation is termed "accessible" and provides support for the assistive technologies used by people with disabilities. For example, accessible documentation provides a text equivalent for each graphic element, such as a picture, flow diagram, or icon. This provision is necessary for users who rely on screen readers to read the documentation.

INTRODUCTION

Last year, the writers at Sun Microsystems went through the process of making their documentation accessible to users with disabilities. Making documentation accessible is the "right thing to do" because more than 54 million Americans experience some activity limitation due to chronic health conditions or impairments. And, accessibility is the law. The strongest legislation in this area is Section 508 of the Federal Rehabilitation Act.

What Is Section 508?

Section 508 requires that electronic and information technology purchased or developed by federal agencies be accessible to people with disabilities. If a product, including documentation, is not accessible, the product might not be considered for government contracts.

Disabilities can be broadly categorized into visual impairments, hearing impairments, and motor limitations. Cognitive disabilities, which range from dyslexia to difficulties with retrieving information from memory, are not addressed directly in this paper. However, creating accessible documentation often enables a person with cognitive disabilities to make effective use of the documentation.

What Is Assistive Technology?

An assistive technology is the hardware and the software that helps people with disabilities use a computer. Following are some common assistive technologies:

- **Screen magnifiers.** Users move the magnifier over the computer screen to enlarge a particular area.
- **Screen readers.** These tools present information from the screen in synthesized speech or Braille.
- **Speech recognition programs.** These programs enable users to control computers by using their voice instead of by using a mouse or a keyboard.
- **On-screen keyboards.** Users select keys from an on-screen keyboard using a variety of specialized pointing devices.

A Note About Our Product

This paper focuses on the steps the writers for the Sun™ ONE Studio product took to make their documentation accessible. The Sun ONE Studio tool is an integrated development environment (IDE) that developers use to create, debug, and deploy applications.

The documentation for the Sun ONE Studio tool (henceforth referred to as the IDE) includes manuals and online help. The manuals are available on the `docs.sun.com` web site and on the product CD. The online help is available with the product and is displayed by using the JavaHelp™ viewer. An open source version of the IDE is the NetBeans™ software and can be downloaded from `http://www.netbeans.org`.

The IDE writers wondered if the accessibility changes made to the documentation would benefit any of the IDE users, so they submitted a survey. Of the 828 users who responded, 23, or 3%, responded that they had a disability that affected their use of the computer. These disabilities ranged from vision impairments, such as color blindness, to motor limitations, such as Carpal Tunnel Syndrome.

A CASE STUDY: MAKING DOCUMENTATION ACCESSIBLE

To make the documentation accessible to users with disabilities, the IDE writers made the following changes to the documentation:

- Provided all documentation in HTML format
- Defined text, color, and spacing in the style sheet
- Added text descriptions to graphic elements
- Ensured that color alone was not used to convey meaning
- Included table summary data and labeled table elements
- Documented the accessibility features of the product

Provide Documentation in HTML Format

The accessibility policy at Sun Microsystems states that all documentation must be available in accessible HTML, accessible PDF, or ASCII format. HTML is the preferred format because HTML can be read by assistive technologies. HTML also provides writers with greater flexibility in formatting their writing.

The IDE writers used various tools for creating the HTML text. Some writers produced documentation using FrameMaker and then converted the documentation into HTML using an in-house tool. A benefit of using an in-house converter is that it adds standard text descriptions for stock graphics, such as the Sun logo and caution and error icons. Other writers used an HTML editor, while still others tagged the HTML themselves.

The writers produced the documentation in HTML 4.0. This version of HTML supports a number of enhancements that enable assistive technologies to extract more information from documents. HTML 4.0 includes new attributes and elements for structuring text, specifying alternate content and descriptions, and facilitating navigation. In addition, HTML 4.0 is the first version of HTML to fully integrate style sheets.

One point that was strongly emphasized was the importance of using correct HTML markup when creating the documents. The writers reviewed all existing documentation and cleaned up any misuse of the HTML tags. For example, some writers had used the `BLOCKQUOTE` element, designed to mark up quotations, for indentation. This use of the tag made the documents less accessible because some assistive technologies cannot decipher text marked up in this way.

The writers also cleaned up their use of the heading tags. Writers should use `H1` for the top-level headings, `H2` for the main divisions of the information within the `H1` heads, and `H3` for lower divisions of the information. Adhering to this hierarchy enables a screen reader to provide an overview of the structure of a page by reading the `H1` and `H2` heads aloud. Proper use of the heading tags also facilitates a user's navigation of the page with a screen reader.

Define Text, Color, and Spacing in the Style Sheet

The size and style of text, the foreground and background color, and the spacing of information can affect the accessibility of documents for users who have vision impairments. For example, low-vision users need to be able to enlarge the type size. Color-blind users need control over text color and background color.

The importance of enabling users to change text they cannot read into a format that they find legible was made clear to the IDE writers during a usability test of the documentation. When testing the online help, the first action a low-vision user took was to increase the text size. The user was able to accomplish this task by overriding the document-defined style sheet with a style sheet of his own.

The following features are defined in the IDE style sheet:

- **Text size and style.** The style sheet specified relative text sizes as opposed to absolute text sizes. This enables the text to grow or shrink as the user issues "text larger" or "text smaller" commands. The text font was also set in the style sheet so that users can easily change the documentation to use their own preferred font.
- **Text and background colors.** The default text color was set to black and the default background color was set to white for maximum contrast.
- **Spacing, alignment, and positioning.** Text indentation, margins, and so on were set in the style sheet. This reduced the need to rely on tables and invisible, single-pixel images to position content.

An early release of the IDE documentation did not use a style sheet and these attributes were defined in individual files. Using a style sheet not only made the documentation more accessible but also improved the appearance of the IDE documentation. Using a style sheet also enabled writers to maintain style changes in one place rather than in each content file.

Within the IDE documentation, users are provided with instructions on how to edit or replace the document-defined style sheet. The style sheet was placed in the top level of the documentation file system so that the style sheet was easy for a user to find and edit.

Users can also read the IDE documentation without a style sheet. This enables assistive technologies to convey information that is displayed in browsers that do not support style sheets.

Add Text Descriptions to Graphic Elements

Section 508, subpart 1194.22(a), requires that a text equivalent be provided for each graphic element, such as a picture, flow diagram, or icon. This provision is necessary because assistive technologies, such as screen readers, cannot interpret graphics. However, if a graphic is accompanied by a text description, the screen reader can read the text in place of the graphic.

The IDE writers followed these general guidelines when providing text equivalents for graphic components:

- For each graphic element, write short alternative text, which must *not* exceed 150 characters in length.
- Optionally, write long alternative text *in addition* to the short alternative text for those graphic elements that cannot be described in 150 characters or less.

Short alternative text was added to graphics by using the `ALT` attribute of the `IMG` element. The IDE documentation includes three types of graphics: functional graphics, screen captures, and placeholder graphics. Following are the guidelines for providing short alternative text for each type of graphic:

- **Functional graphics.** Use the alternative text to describe the function that the graphic performs. For example, the IDE online help uses a right arrow graphic as a shorthand way to refer to the opening of menu items. Following is an example:

  From the main window, choose Debug > New Breakpoint.

  The HTML markup for this sentence is:

  ```
  From the main window, choose Debug
  <IMG SRC="/images/arrow.gif" ALT="and
  choose"> New Breakpoint.
  ```

  In this case, the alternative text describes the action the graphic performs (“and choose”) as opposed to explaining the appearance of the graphic (“right arrow”).

- **Screen captures.** Use the alternative text to identify the graphic and describe the visual elements that are relevant to the user action at the current point in the document. Following are two examples:

  ```
  <IMG SRC="EmphasizedTargetIcon.gif"
  ALT="Node icon for Ant targets with
  description">
  ```

  ```
  <IMG SRC="FinishSessionDialog.gif"
  ALT="Finish Debugging Sessions dialog
  box showing sessions to terminate
  when you stop the debugger. Buttons
  are OK and Cancel.">
  ```

- **Placeholder graphics.** Use an empty `ALT` attribute for images that do not convey important information or convey redundant information. These include images used for spacing and decorative graphics. For example:

  ```
  <IMG SRC="placeholder.gif" ALT="">
  ```

Do not break the alternative text across lines in the source code. The examples shown here cannot show this, as the column width of this document is too narrow.

Long alternative text can be added to a graphic by using the `LONGDESC` attribute of the `IMG` tag. In all cases, the IDE writers found that the short alternative text was sufficient for providing a text equivalent for a graphic. Therefore, no long alternative text was provided in the documentation.

Ensure That Color Alone Is Not Used to Convey Meaning

Section 508, subpart 1194.22(c), prohibits using color as the only way to convey important information. This requirement is especially important for color-blind users who have trouble distinguishing between combinations of colors.

The main area in which this requirement affected the IDE documentation was in the representation of the glossary links in the online help. The JavaHelp viewer does not support the underlining of glossary links. Before Section 508, the writers identified these links using only the color blue. After Section 508, the writers changed these links so that the text is in bold, italic, and blue.

Include Table Summary Data and Label Table Elements

Section 508, subparts (g) and (h), require data tables to include markup that enables assistive technologies to convey the structure of the table. The IDE writers followed these guidelines when creating data tables:

- Summarize the table contents by including a table caption, by introducing the table in the surrounding text, or by using the `SUMMARY` attribute of the `TABLE` element.
- Include column headers in the table. Provide row headers if these headers make the table easier to navigate.
- Use markup to associate data cells with header cells.
- Use relative, as opposed to absolute, widths and heights in defining table cells. Specifying a fixed size for a table might introduce formatting difficulties if the user resizes the window in which the table is displayed.

Following is an example of a data table introduced in the surrounding text and the HTML markup for the table. The `SCOPE` attribute is set to `COL` in the table header element `TH` to indicate the cell is a column header. The WIDTH attribute is set to 30%.

```
The following table lists the commands
to start a debugging session.

<TABLE BORDER="1">
<TR>
<TH SCOPE="COL" WIDTH="30%">Command
</TH>
<TH SCOPE="COL">Description
</TH>
</TR>
<TR>
<TD>Run in Debugger
</TD>
<TD>Starts a debugging session.
</TD>
</TR>
<TR>
<TD>Attach
</TD>
<TD>Attaches the debugger to a process
that is already running.
</TD>
</TR>
</TABLE>
```

This table appears as shown here:

The following table lists the commands to start a debugging session.

Command	Description
Run in Debugger	Starts a debugging session.
Attach	Attaches the debugger to a process that is already running.

The guidelines for providing markup for tables are geared toward data tables and not tables used to control layout. Layout tables are often used for two-column output and code boxes. The IDE writers used layout tables to indent the links in the See Also lists in the online help. For these and other layout tables, no additional markup was provided.

Because layout tables can cause problems for some older screen readers, it is better to use a style sheet for layout whenever possible. Most layout tables in the IDE online help were designed prior to Section 508 and prior to using style sheets.

Document the Accessibility Features of the Product

The IDE writers added a section to both the manuals and online help to describe the accessibility features of the product. This documentation addressed the requirements in Section 508, subpart 1194.41(b). The IDE documentation describes the following:

- How to use keyboard shortcuts and mnemonics
- How to use the code completion feature of the editor
- How to change font and color in the IDE
- How to customize menus and toolbars
- How to use an alternative style sheet to format files

There was considerable debate among the writers as to how to make this information available to the user. Should users be able to access the information from the Welcome screen in the product? Should an Accessibility item be added to the Help menu? The final decision was to include the information in the documentation and place it after the description of the main product features. Users can access the information from the table of contents, the index, and by searching.

Documenting the accessibility features of the product does not necessarily require additional work. Documentation that completely describes the user interface often already includes this information.

ASSESSING THE DOCUMENTATION FOR ACCESSIBILITY

Once the writers changed the documentation to meet accessibility requirements, they were eager to check their work. The IDE writers ran the documentation through the Bobby Web Page Accessibility Checker. Bobby outputs a message for each accessibility problem on a page. For example, Bobby indicates when the alternative text for a graphic is missing, when color is misused, and when a table is not marked up properly.

The IDE writers did further tests on the documentation by testing the documentation with the JAWS 4.0 screen reader. This testing enabled the writers to hear how a screen reader would read the documentation aloud, especially the alternative text for graphics and the text in data tables and layout tables. Another way to test the alternative text for graphics is to read aloud the alternative text and the figure caption.

This testing, however, did not guarantee that the documentation was fully accessible. Although Bobby can indicate when the alternative text for a graphic is missing, the tool cannot guarantee that the alternative text is the appropriate text. In addition, JAWS is only one of many

different screen readers and one of many different types of assistive technologies.

THIRD-PARTY DOCUMENTATION AND ACCESSIBILITY

One of the accessibility "gotchas" that the IDE writers ran into involved the third-party documentation provided with the product. The writers did not realize until late into the product release cycle that the third-party documents did not meet accessibility requirements. The writers hurried to add the alternative text to the graphics and the markup to tables before the deadline. To prevent this problem from happening again, the writers now provide suppliers of all third-party documents with a copy of the *Sun ONE Studio IDE Accessibility Style Guide*.

WHAT'S NEXT: HELP FILES ON THE WEB AND MORE

The IDE writers plan to provide a browser-ready version of the online help in addition to the JavaHelp version. Users can view this help in their own browser (such as Netscape Navigator or Internet Explorer) as opposed to the JavaHelp viewer. Creating this version of the documentation requires the following changes to the JavaHelp files:

- Convert the `OBJECT` tag to hypertext links. The `OBJECT` tag is specific to the JavaHelp software and is used to create a pop-up window.
- Convert the XML table of contents and index to HTML.
- Add links to the table of contents and index at the top of each file.

The writers also plan to perform a usability test to evaluate the quality of the text equivalents for the graphic elements. Writers must also be aware of accessibility requirements as they develop new content or update existing documentation.

ADDITIONAL TIPS

This paper described the steps that the IDE writers took to make their documentation meet accessibility requirements. The W3C Web Accessibility Initiative Quick Tips Reference Card has additional tips that might apply to your documentation. The card is available at `http://www.w3.org/WAI/References/QuickTips/`.

CONCLUSION

Documentation must take into account the needs of people with disabilities. Such documentation is termed "accessible" and provides support for the assistive technologies used by people with disabilities. Creating accessible documentation is largely a matter of applying the principles of good design:

- Use correct HTML markup
- Define text size and font, foreground and background color, and spacing and alignment in the style sheet
- Provide text equivalents for graphic elements
- Ensure that color alone is not used to convey meaning
- Include table summary data and label table elements
- Document the accessibility features of the product

Good design benefits not only users with disabilities, but all users.

REFERENCES

(1) Bobby. Retrieved July 29, 2002 from the World Wide Web: `http://bobby.watchfire.com/bobby/html/en/index.jsp`

(2) NetBeans IDE. Retrieved January 30, 2003 from the World Wide Web: `http://www.netbeans.org`

(3) JAWS for Windows. Retrieved January 30, 2003 from the World Wide Web: `http://www.freedomscientific.com/fs_downloads/jaws.asp`

(4) Section 508. Retrieved July 29, 2002 from the World Wide Web: `http://www.section508.gov`

(5) Sun Microsystems Accessibility Program. Retrieved July 29, 2002 from the World Wide Web: `http://www.sun.com/access/`

(6) WAI Quick Tips Reference Card. Retrieved January 30, 2003 from the World Wide Web: `http://www.w3.org/WAI/References/QuickTips/`

(7) Web Content Accessibility Guidelines 1.0. Retrieved July 29, 2002 from the World Wide Web: `http://www.w3.org/TR/WAI-WEBCONTENT/`

Gail B. Chappell
Senior Technical Writer
Sun Microsystems
4150 Network Circle M/S MPK16-305
Santa Clara, CA 95054

Gail Chappell has worked as a technical writer at Sun Microsystems for over 16 years. She has a B.A. in Journalism and a B.A. in German from the University of California, Berkeley.