A process model for creating accessible end-user technical documentation

Richard D. Herring, Ph.D

STC Annual– May 2005

IP Telephony | Contact Centers | Unified Communication | Services

Presentation Information

- **Session: Creating accessible documents (UID 11L)**
- **Format: Paper**
- **Date: Wednesday, May 11th, 2005**
- **Time: 2:00 pm – 3:30 pm (PST)**
- **Setting: Society for Technical Communication, 52nd Annual Meeting, Seattle, WA (stc.org)**
- **Note: These slides supplement the paper published in the 52nd Annual Proceedings**

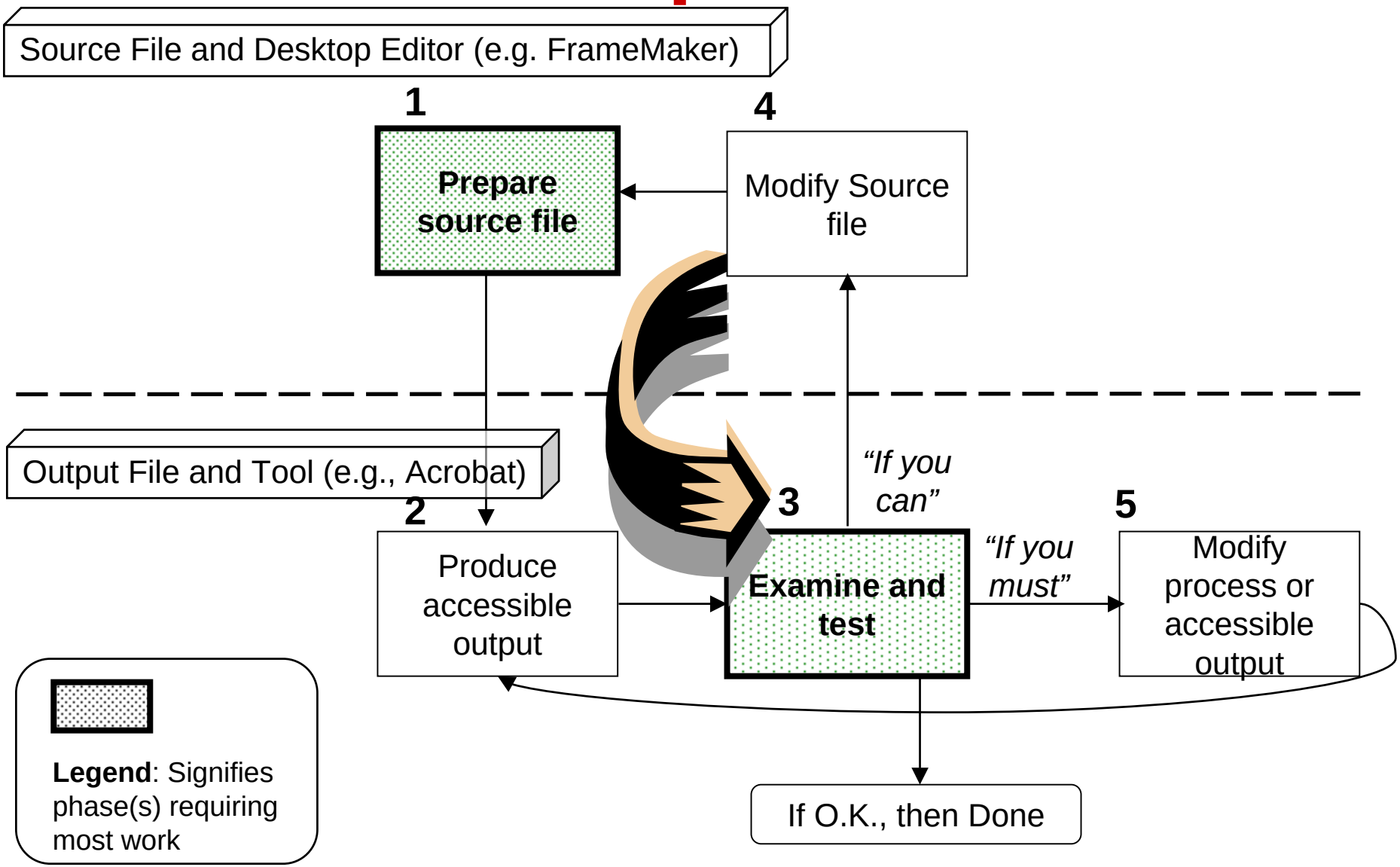
Goals of the presentation

- **Introduce a Five-Part model for creating accessible documents**
- **Illustrate Step 1 - subtleties in creating “text equivalents”**
- **Detail Step 3 - three types of testing**
- **Example … STC Proceedings**

Working definition of an accessible document

- A document whose electronic form can be adequately read by an electronic screen reader
- Definition modified from James Thatcher (see http://jimthatcher.com/)

Accessible Output Process



1. Prepare source file

- **Adding “text equivalents” that represent a visual experience**
- **Must apply special editing**
- **Must add “affordances” for differently-abled readers**

Adding Text Equivalents - Accessible PDF

Next are located under the softkeys. See Figure 4.¶

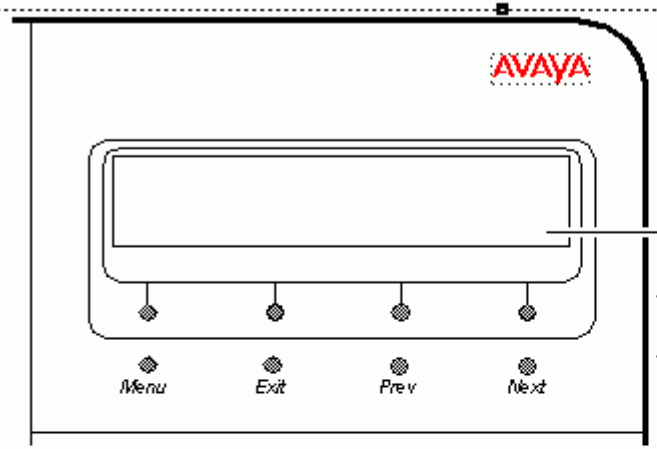
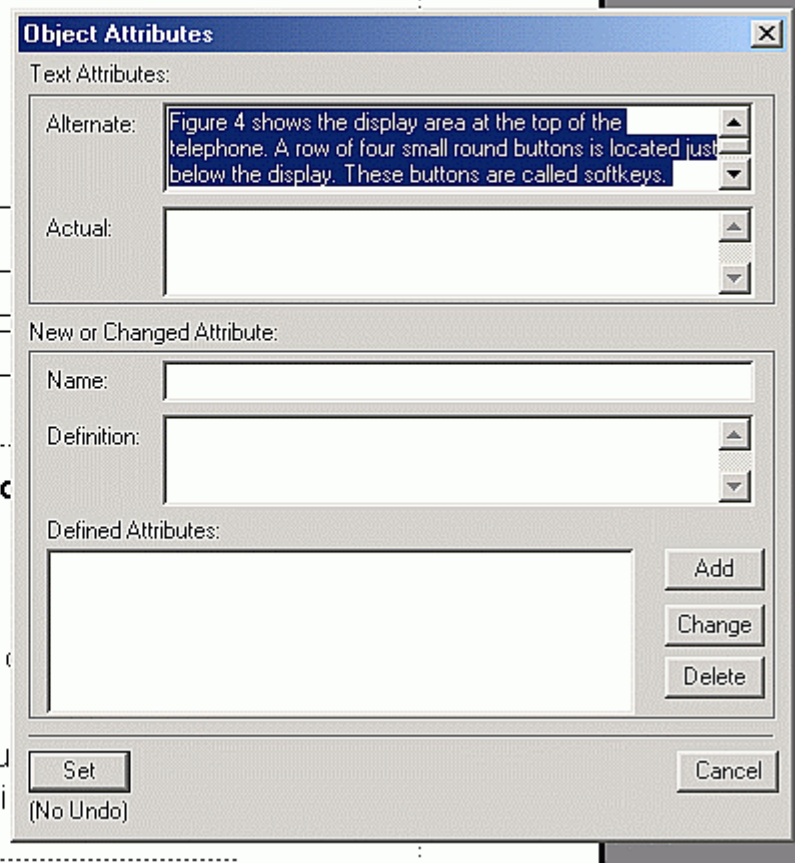


FIGURE 4) ¶The Display, Softkeys, and Display Co

Using the Softkey Feature Menus ¶

There are three separate softkey feature menus. Each
you to select from four different features.¶

You can enter Softkey Mode (and view the softkey featu
the display control button labeled **Menu**. The following i
softkey feature menu.¶

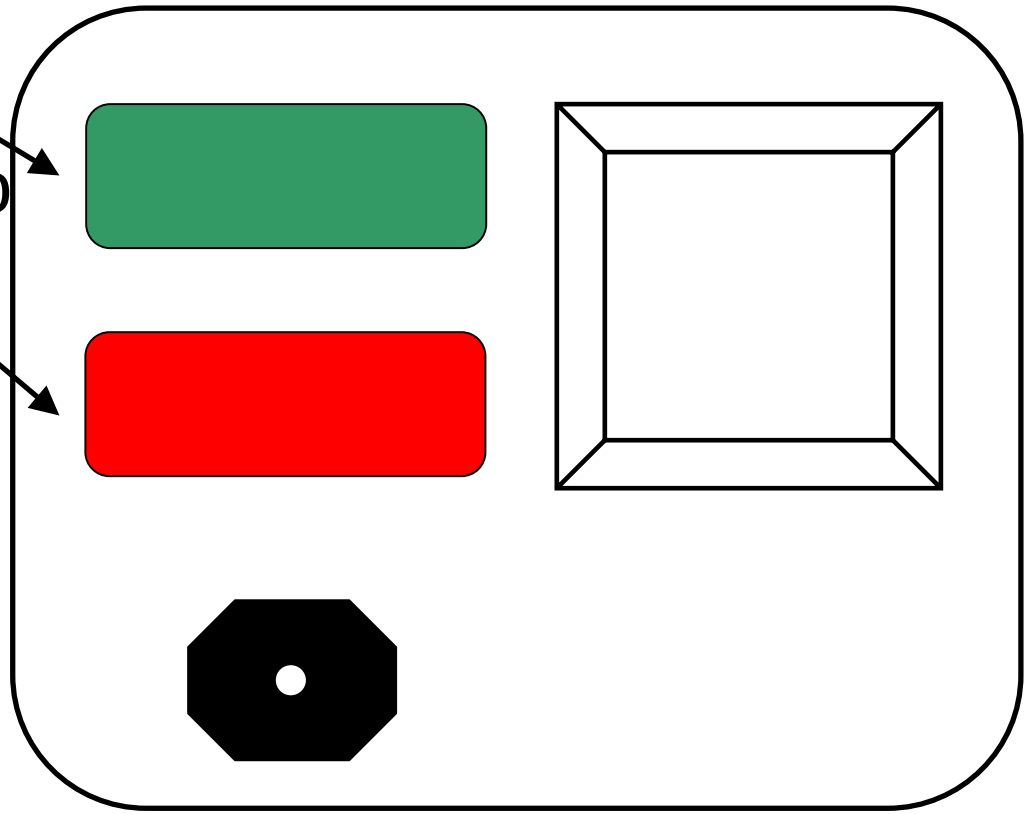


Editing Example: “Press the green button then submit”

Note: May appear as shades of gray to

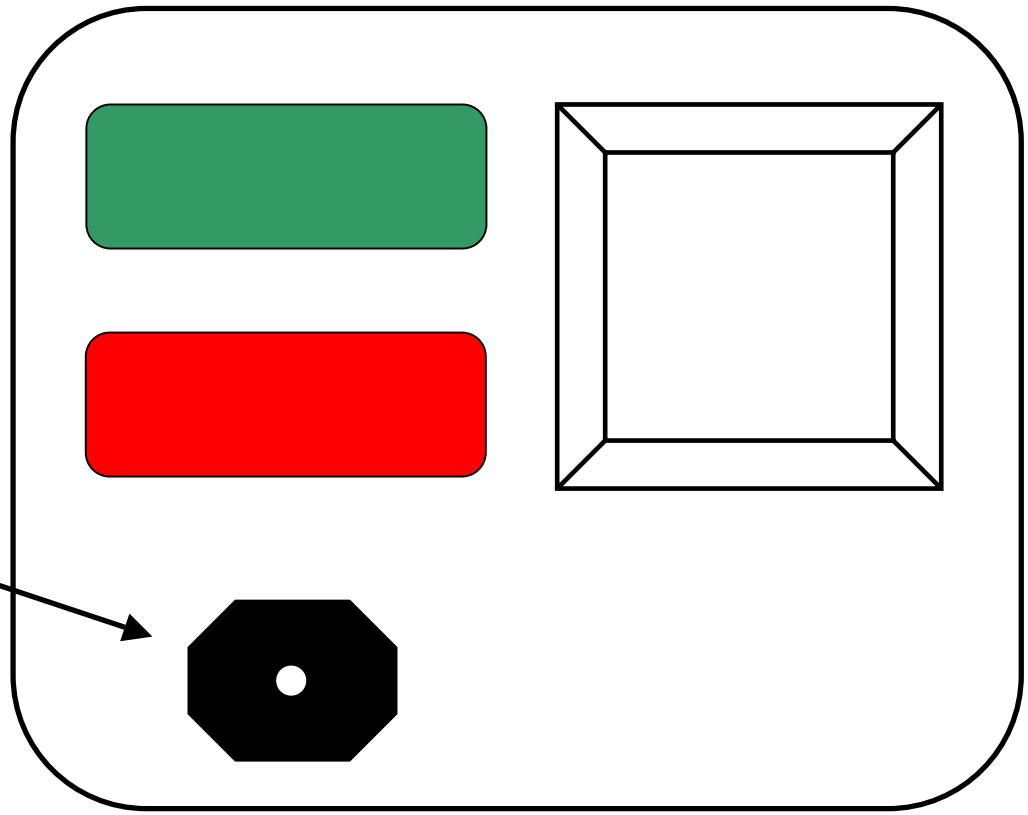
-People with color-blindness

-Monochrome printers



Better: “Press the topmost rectangular button above the octagonal button …

Note: Raised nub on button surface for tactile cue



Editorial suggestions for text equivalents

- Describe graphic “value-add”
- Provide a logical overall orientation … part/whole; spatial organization; etc.
- Provide physical reference points … above “x”; at the Two o'clock Position; etc.
- Provide tactile and spatial characteristics

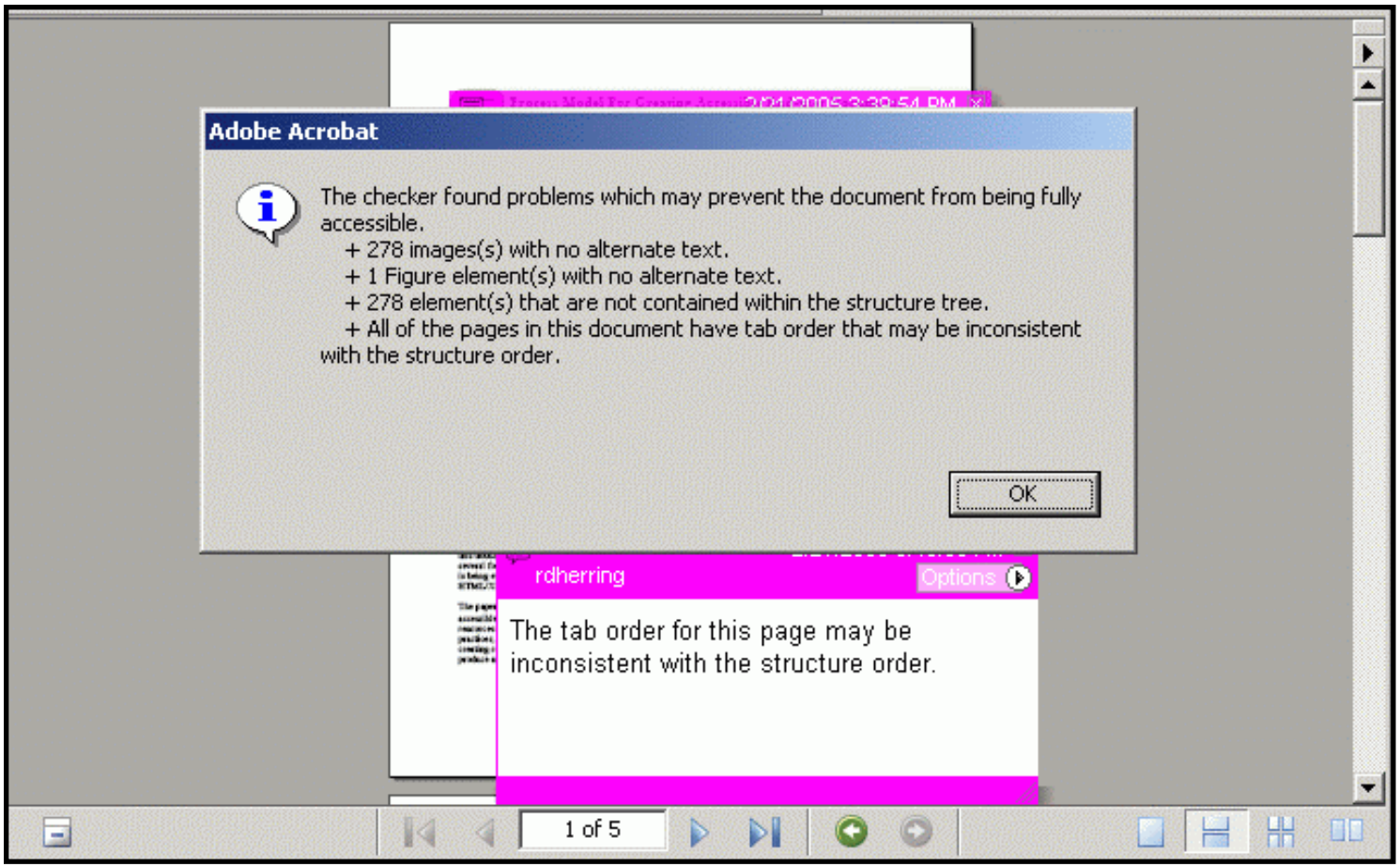
2. Produce accessible output from the source file

- Complete the production process
- Use a “Save as …” option to render source file to another form

3. Examine and test accessible output

- Use automated tests
- Use manual (i.e. visual) tests
- Test with special populations

Automated test: Initial results



Manual test: Linear Reading Order (.txt file output)

Table 4-ASCII text file inappropriately rendered: missing inline graphics shown as "♦"

Note: If your telephone is connected to DEFINITY Release 7.1 or a later release and if there is only one call on hold at your telephone, you can transfer the call or initiate a conference call by pressing♦ or♦ without first returning to the held call.

Table 5-Replacing inline graphics with alternate text (equivalent text shown in a different color and font for emphasis)

Note: If your telephone is connected to DEFINITY Release 7.1 or a later release and if there is only one call on hold at your telephone, you can transfer the call or initiate a conference call by pressing the oval Transfer button, which is the middle button in the first row of feature buttons above the dial pad, or *the oval Conference button, which is the last button on the right, in the first row of feature buttons above the dial pad,* without first returning to the held call.

Test with your target population – possibly differently-abled

4. Modify source file and produce accessible output again

- If at first you don’t succeed … change the raw materials in the process
- This step more desirable to take than step #5

Revisit: Adding text alternatives to source

Next are located under the softkeys. See Figure 4.¶

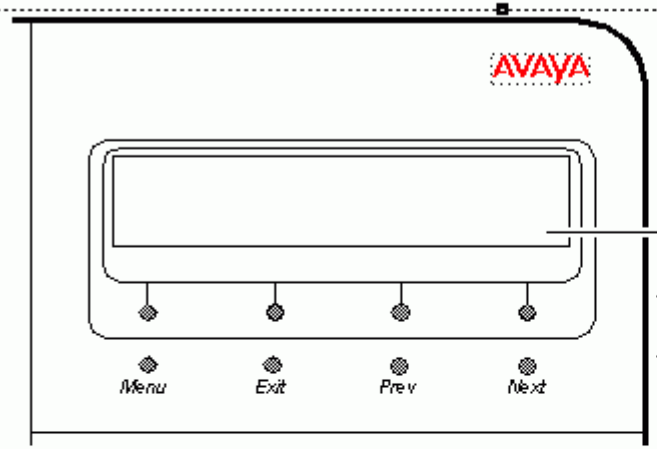
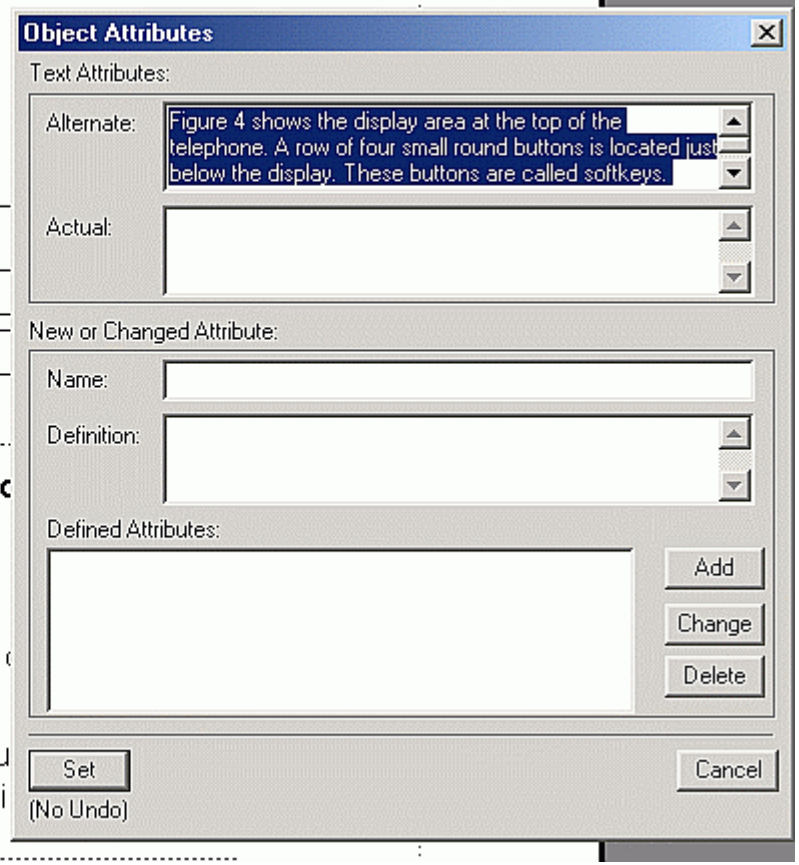


FIGURE 4) ¶The Display, Softkeys, and Display Co

Using the Softkey Feature Menus ¶

There are three separate softkey feature menus. Each
you to select from four different features.¶

You can enter Softkey Mode (and view the softkey featu
the display control button labeled **Menu**. The following i
softkey feature menu.¶



5. Modify production process or accessible output again

- **If at *second* you don't succeed … change the process or its resulting product**
- **It is more desirable to take step #4**

Example – Print version of this session's paper

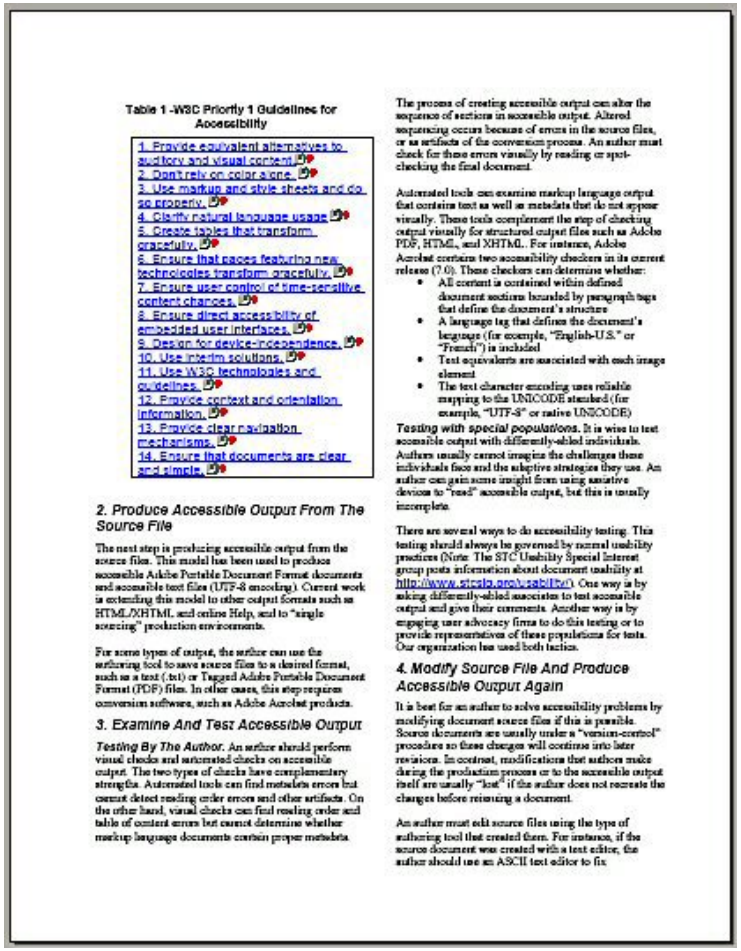


Table 1 -W3C Priority 1 Guidelines for Accessibility

1. Provide equivalent alternatives to auditory and visual content.
2. Don't rely on color alone.
3. Use markup and style sheets and do so properly.
4. Clarify natural language usage.
5. Create tables that transform gracefully.
6. Ensure that pages featuring new technologies transform gracefully.
7. Ensure user control of time-sensitive content changes.
8. Ensure direct accessibility of embedded user interfaces.
9. Design for device-independence.
10. Use interim solutions.
11. Use W3C technologies and guidelines.
12. Provide context and orientation information.
13. Provide clear navigation mechanisms.
14. Ensure that documents are clear and simple.

2. Produce Accessible Output From The Source File

The next step is producing accessible output from the source files. This model has been used to produce accessible Adobe Portable Document Format documents and accessible text files (UTF-8 encoding). Current work is extending this model to other output formats such as HTML/XHTML and online Help, and to "single sourcing" production environments.

For some types of output, the author can use the authoring tool to save source files to a desired format, such as a text (.txt) or Tagged Adobe Portable Document Format (PDF) files. In other cases, this step requires conversion software, such as Adobe Acrobat products.

3. Examine And Test Accessible Output

Testing By The Author. An author should perform visual checks and automated checks on accessible output. The two types of checks have complementary strengths. Automated tools can find metadata errors but cannot detect reading order errors and other artifacts. On the other hand, visual checks can find reading order and table of content errors but cannot determine whether markup language documents contain proper metadata.

The process of creating accessible output can alter the sequence of sections in accessible output. Altered sequencing occurs because of errors in the source files, or as artifacts of the conversion process. An author must check for these errors visually by reading or spot-checking the final document.

Automated tools can examine markup language output that contains text as well as metadata that do not appear visually. These tools complement the step of checking output visually for structured output files such as Adobe PDF, HTML, and XHTML. For instance, Adobe Acrobat contains two accessibility checkers in its current release (7.0). These checkers can determine whether:

- All content is contained within defined document sections bounded by paragraph tags that define the document's structure
- A language tag that defines the document's language (for example, "English-U.S." or "French") is included
- Text equivalents are associated with each image element
- The text character encoding uses reliable mapping to the UNICODE standard (for example, "UTF-8" or native UNICODE)

Testing with special populations. It is wise to test accessible output with differently-abled individuals. Authors usually cannot imagine the challenges these individuals face and the adaptive strategies they use. An author can gain some insight from using assistive devices to "read" accessible output, but this is usually incomplete.

There are several ways to do accessibility testing. This testing should always be governed by normal usability practices (Note: The STC Usability Special Interest group posts information about document usability at http://www.stcsig.org/usability). One way is by asking differently-abled associates to test accessible output and give their comments. Another way is by engaging user advocacy firms to do this testing or to provide representatives of these populations for tests. Our organization has used both tactics.

4. Modify Source File And Produce Accessible Output Again

It is best for an author to solve accessibility problems by modifying document source files if this is possible. Source documents are usually under a "version-control" procedure so these changes will continue into later revisions. In contrast, modifications that authors make during the production process or to the accessible output itself are usually "lost"[4] if the author does not recreate the changes before reissuing a document.

An author must edit source files using the type of authoring tool that created them. For instance, if the source document was created with a text editor, the author should use an ASCII text editor to fix

First electronic submission – whole page is a “table”

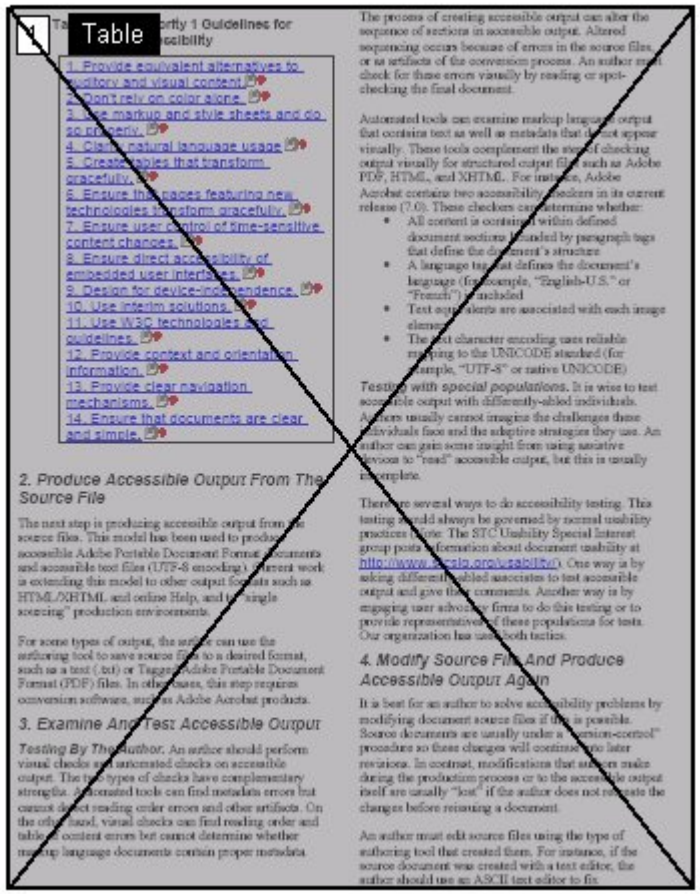
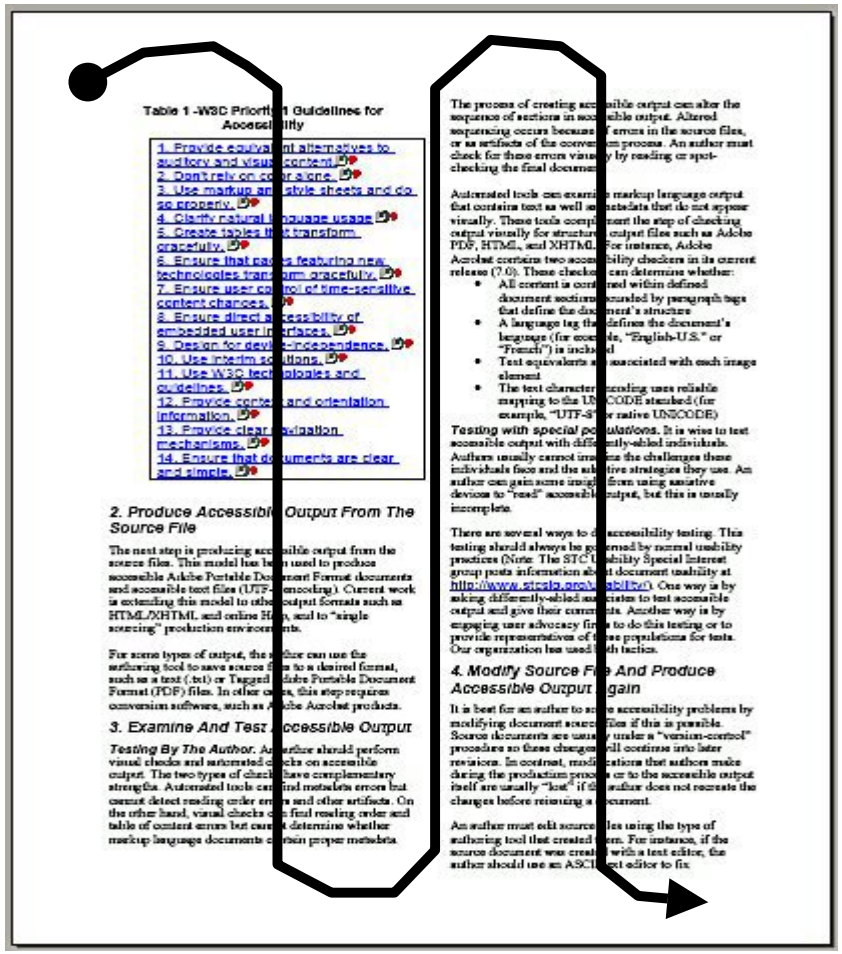
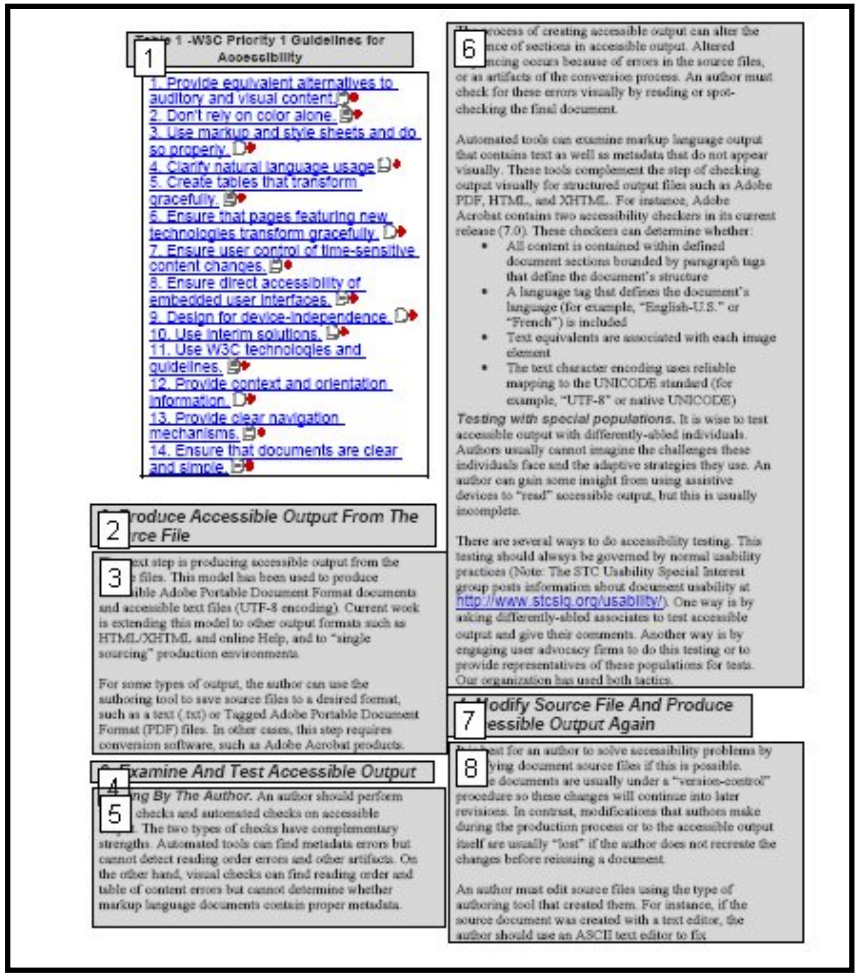


Table Text flow – is still a linearized reading order



Allows “text reflow” to handheld machine

But it misses author's intended structure – articulated



1 -WSC Priority 1 Guidelines for Accessibility

1. Provide equivalent alternatives to auditory and visual content.
2. Don't rely on color alone.
3. Use markup and style sheets and do so properly.
4. Clarify natural language usage.
5. Create tables that transform gracefully.
6. Ensure that pages featuring new technologies transform gracefully.
7. Ensure user control of time-sensitive content changes.
8. Ensure direct accessibility of embedded user interfaces.
9. Design for device-independence.
10. Use interim solutions.
11. Use W3C technologies and guidelines.
12. Provide context and orientation information.
13. Provide clear navigation mechanisms.
14. Ensure that documents are clear and simple.

2 *roduce Accessible Output From The rce File*

3 xt step is producing accessible output from the files. This model has been used to produce ible Adobe Portable Document Format documents and accessible text files (UTF-8 encoding). Current work is extending this model to other output formats such as HTML/XHTML and online Help, and to "single sourcing" production environments.

For some types of output, the author can use the authoring tool to save source files to a desired format, such as a text (.txt) or Tagged Adobe Portable Document Format (PDF) files. In other cases, this step requires conversion software, such as Adobe Acrobat products.

4 *xamine And Test Accessible Output*

5 *ng By The Author.* An author should perform checks and automated checks on accessible The two types of checks have complementary strengths. Automated tools can find metadata errors but cannot detect reading order errors and other artifacts. On the other hand, visual checks can find reading order and table of content errors but cannot determine whether markup language documents contain proper metadata.

6 rocess of creating accessible output can alter the nce of sections in accessible output. Altered ncing occurs because of errors in the source files, or in artifacts of the conversion process. An author must check for these errors visually by reading or spot-checking the final document.

Automated tools can examine markup language output that contains text as well as metadata that do not appear visually. These tools complement the step of checking output visually for structured output files such as Adobe PDF, HTML, and XHTML. For instance, Adobe Acrobat contains two accessibility checkers in its current release (7.0). These checkers can determine whether:

- All content is contained within defined document sections bounded by paragraph tags that define the document's structure
- A language tag that defines the document's language (for example, "English-U.S." or "French") is included
- Text equivalents are associated with each image element
- The text character encoding uses reliable mapping to the UNICODE standard (for example, "UTF-8" or native UNICODE)

Testing with special populations. It is wise to test accessible output with differently-abled individuals. Authors usually cannot imagine the challenges these individuals face and the adaptive strategies they use. An author can gain some insight from using assistive devices to "read" accessible output, but this is usually incomplete.

There are several ways to do accessibility testing. This testing should always be governed by normal usability practices (Note: The STC Usability Special Interest group posts information about document usability at http://www.stcsig.org/usability/). One way is by asking differently-abled associates to test accessible output and give their comments. Another way is by engaging user advocacy firms to do this testing or to provide representatives of these populations for tests. Our organization has used both tactics.

7 *odify Source File And Produce essible Output Again*

8 st for an author to solve accessibility problems by ying document source files if this is possible. e documents are usually under a "version-control" procedure so these changes will continue into later revisions. In contrast, modifications that authors make during the production process or to the accessible output itself are usually "lost" if the author does not recreate the changes before reissuing a document.

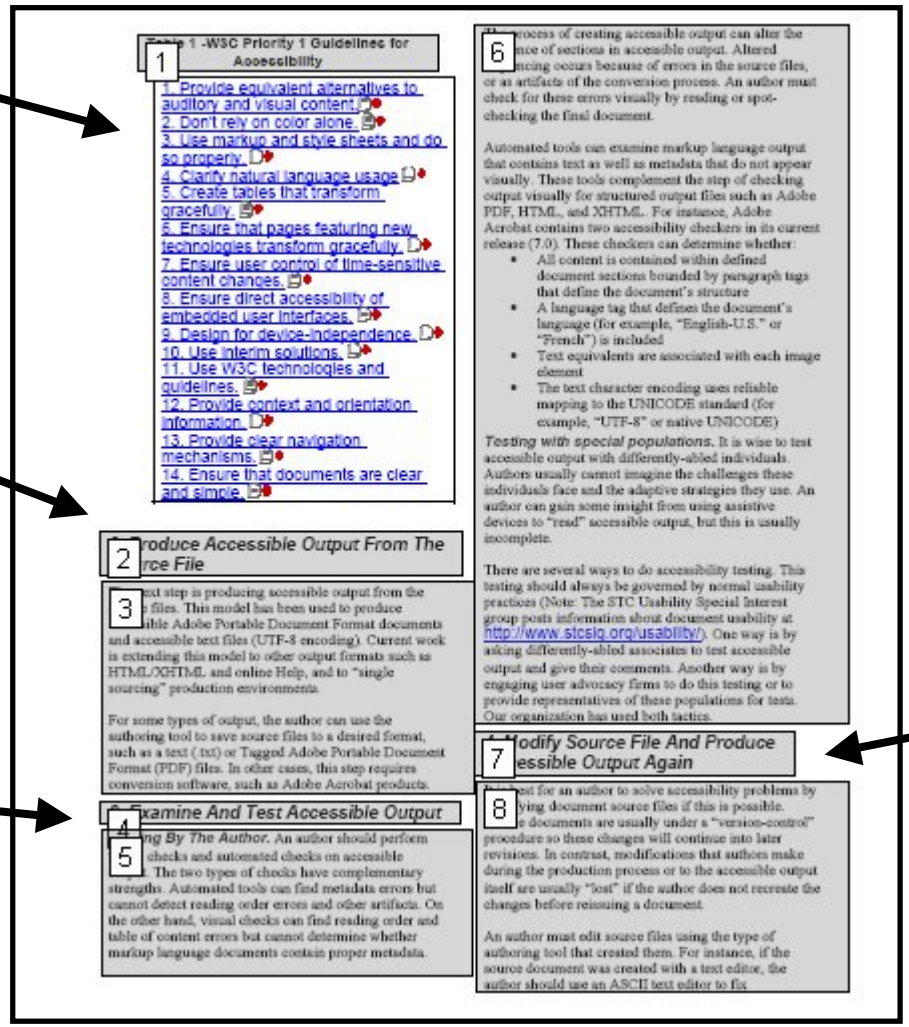
An author must edit source files using the type of authoring tool that created them. For instance, if the source document was created with a text editor, the author should use an ASCII text editor to fix

Final submission – modified

Table

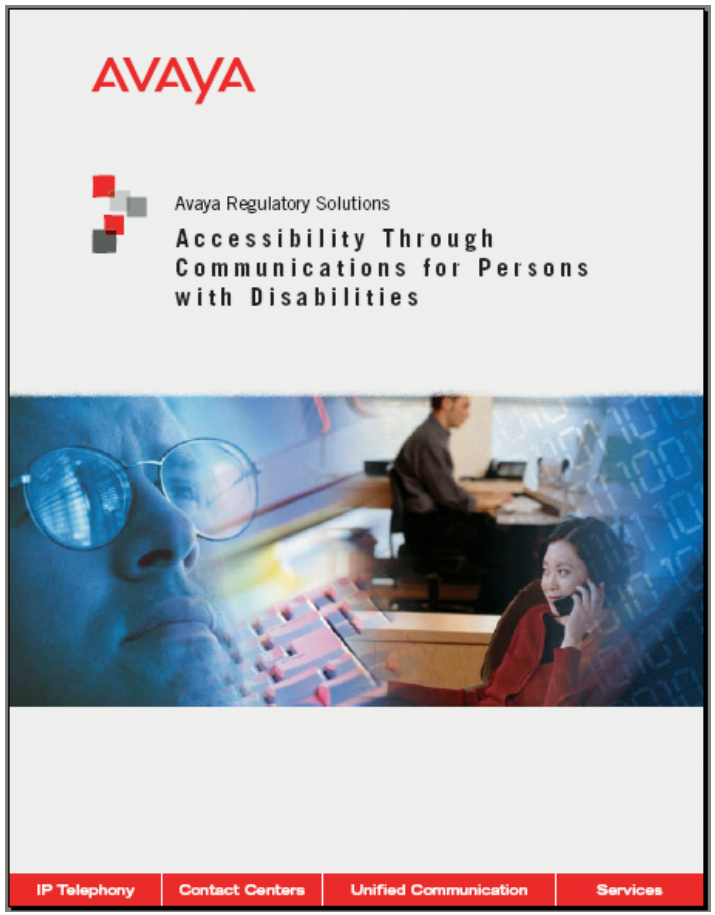
Heading Level 2

Heading Level 2



Heading Level 2

What if … content could speak for itself?



Accessible Solutions Brochure

http://www1.avaya.com/enterprise/508/

Compatible with Adobe Acrobat 6.0 "Read Aloud" Feature

Click speaker on .PPT version for audio .. Or surf to web address above

Conclusions

- Introduce a Five-Part model for creating accessible documents
- Illustrate Step 1 - subtleties in creating “text equivalents”
- Detail Step 3 - three types of testing
- Example … STC Proceedings

Acknowledgements

- Donna Fioto, Dr. Paul Michaelis, Bill West
- Greg Pisocky (Adobe), Debra Ruh (TecAccess)
- Elise Amorosi, Chris Foard, A. Ken Herron, Laurie Kraus, Bobbi Montoya & … others
- Lou Lombardi, Paul Newland
- And others …

Thank you!

- Questions and comments welcome

Richard D. Herring, Avaya

Performance Communications Group (rdherring@avaya.com)

(alternate email:kg2pu@arrl.net)