



Authoring for
Accessibility and Reflow
in Adobe FrameMaker
and Adobe Acrobat



Module 1: Introduction to Accessibility Issues



Introduction

This module introduces you to the challenges of creating accessible electronic documents for individuals with disabilities. Creating accessible documents requires that we rethink the way we present information. It requires using software that produces documents that can preserve the intended content ordering and provide alternate descriptions for visuals such as pie charts, tables, and graphics. Adobe Tagged PDF provides the functionality needed to support document accessibility.

During this course, you will learn guidelines to produce optimal Adobe Tagged PDF files from a particular authoring application, such as Adobe InDesign or Microsoft Word. The hands-on exercises included in this training course allow you to troubleshoot accessibility problems in a document so that the content sequence of the original is retained when the document is accessed through an alternate device such as a screen reader. You will also learn how to further enhance a Tagged PDF file using Adobe Acrobat's accessibility tools.

In addition to this course, you can also refer to the Web sites and other resources listed in the [References](#) section in each module for further material.

Learning Objectives

At the end of this module, you will be able to:

- Describe aspects of document design that can cause accessibility problems
- Describe the difficulties in presenting information that is normally in graphics and tables to someone who is visually impaired
- Understand that Tagged PDF provides structure information that is used by screen readers in presenting information to a visually-impaired person
- Understand the process for creating and troubleshooting accessible documents

References

- <http://access.adobe.com>
- <http://www.adobe.com/products/acrobat/solutionsacc.html>

Product Information

- *How To Create Accessible Adobe PDF Files*
 - PDF Version: <http://www.adobe.com/products/acrobat/pdfs/accessbooklet.pdf>
 - HTML Version: <http://access.adobe.com/booklet1.html>
- *Advanced Techniques for Creating Accessible Adobe PDF Files*
 - <http://www.adobe.com/products/acrobat/pdfs/CreateAccessibleAdvanced.pdf>
- White Paper: *Enhancing the accessibility of the Web with Adobe Acrobat software*

- PDF Version:
<http://www.adobe.com/products/acrobat/pdfs/accesswhitepaper.pdf>
- HTML Version: <http://access.adobe.com/whitepaper1.html>
- Section 508 Analysis:
http://www.adobe.com/products/acrobat/pdfs/Sect_5085.pdf
- *Acrobat 5.0 and Accessibility Frequently Asked Questions*
<http://www.adobe.com/products/acrobat/pdfs/Acro5AccessFAQ.pdf>
- Section 508 Press Release
<http://www.adobe.com/aboutadobe/pressroom/pressreleases/200106/20010621508.html>
- FOSE Press Release (discussing Acrobat 5.0 as it relates to Government and Accessibility)
<http://www.adobe.com/aboutadobe/pressroom/pressreleases/200103/20010320fose.html>

Product Demonstrations

- Adobe Acrobat 5.0 and Accessibility Video Series
 - Working with Microsoft Office 2000 Files
<http://www.adobe.com/products/acrobat/movie1.html>
 - Working with Existing PDF files
<http://www.adobe.com/products/acrobat/movie2.html>
 - Working with Forms
<http://www.adobe.com/products/acrobat/movie3.html>
 - Usability Enhancements of Acrobat 5.0
<http://www.adobe.com/products/acrobat/movie4.html>
 - Working with Screen Readers
<http://www.adobe.com/products/acrobat/movie5.html>

Downloads

- Acrobat 5.05 Update (you must have Acrobat 5.0 installed to use this)
 - <http://www.adobe.com/products/acrobat/update.html>
- Acrobat Reader 5.0 Download (the version that includes accessibility)
 - standard presentation page:
<http://www.adobe.com/products/acrobat/readstep.html>
 - text-only page:
<http://www.adobe.com/products/acrobat/alternate.html>
- Additional Adobe product downloads for accessibility
<http://access.adobe.com/downloads.html>

Section 508 Information

- <http://www.access-board.gov/sec508/guide>

- <http://www.section508.gov/>
- <http://www.access-board.gov/news/508-final.htm>
- <http://www.itpolicy.gsa.gov/cita/fap.htm>

Vendor Information

- Freedom Scientific Jaws (<http://www.freedomscientific.com>)
- GW Micro WindowEyes (<http://www.gwmicro.com>)
- Dolphin Oceanic HAL (<http://www.dolphinoceanic.com>)
- <http://www-3.ibm.com/able/overview.html>
- <http://www.microsoft.com/enable>

Contents

Topics	Exercises
Section 508 and Electronic Documents	
How Acrobat Helps With Accessibility	
Accessibility Design Considerations	

File to Download for Exercises in this Module

None



Section 508 and Electronic Documents

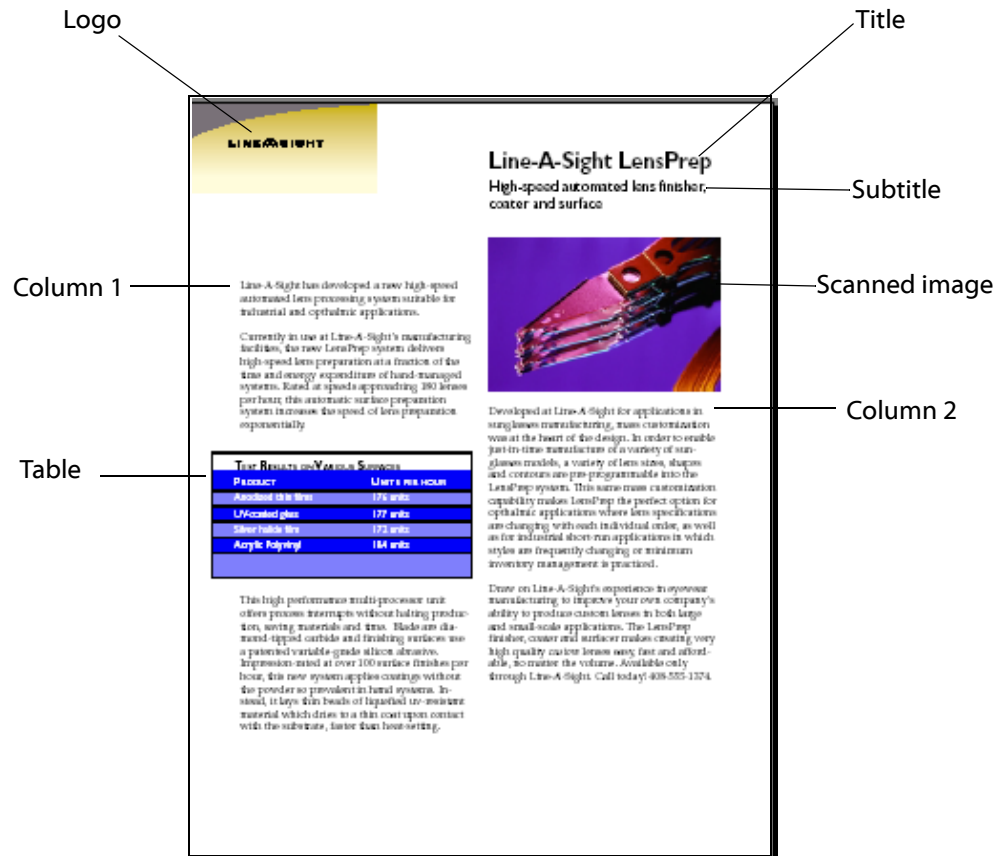
The World Wide Web has revolutionized the flow of information to people around the globe. However, not everyone has been able to partake in this ready access to information. In particular, individuals who are blind, have low visual acuity, or have motor impairment can find interacting with computer technologies challenging and frustrating.

Instructor Notes: Mention that Section 508 covers a broad range of disabilities in many situations; access to electronic documents is just one area. Because documents are by nature visual, the focus of this course is toward providing access to electronic documents for blind and visually impaired persons.

The Architectural and Transportation Barriers Compliance Board (Access Board) issued accessibility standards for electronic and information technology on December 21, 2000. This document is known as Section 508 and requires Federal agencies' electronic and information technology is accessible to people with disabilities. Information technology also encompasses electronic documents and the software used to read them.

Accessibility should be a relatively easy, low-effort addition to the documentation production process. This can be true provided that you consider accessibility during the document design phase. Most documents are geared towards sighted individuals. Layouts of information that interweave graphics and text with the intention of being read in a particular reading order can result in a confused presentation from the screen reader. A screen reader is a voice-synthesized program that converts the text of a document to an audible stream. People who rely on screen magnifiers to read information can only view a small portion of the document at a time and may have problems navigating a document with an artistic layout. Sighted people can easily grasp trends from data presented in graphs and tables, but that visual context is completely lost for a visually-impaired person.

The following document illustrates some of these points. Should the screen reader ignore the logo? Where should it start reading? Does the picture convey something that is important for non-sighted viewers to understand? If so, what is the best way to explain this in text? How can the information presented in the table be summarized effectively? In order for a vision-impaired reader to make sense of this document, we must somehow convey the structure as well as the content of the document. For example, what order should information be read. What should be said about the pictures and tables?



Proper presentation of information depends on several factors. Encoding the proper, or logical, reading order for a document is one of the most fundamental steps towards accessibility. This is especially critical for documents with columnated formats or where there are several distinct text blocks. Designing a document for accessibility means rethinking the message that you intend to convey with visuals. What is a table of data supposed to tell you if you cannot easily reference the row and column headings to give you context for the data contained within it? The bottom line is that the simpler a document's format and layout, the easier it will be to create a truly accessible document.

[Instructor Notes: It is possible to make visually complex documents accessible, but it can take a great deal of work depending on the complexity of the document and the authoring software.](#)

As equally important as the document is the software that a person with disabilities uses to access the document. The availability of the following features greatly enhances this user's interaction with software:

- Ability to process the logical read order of a document.

Encoding the logical read order in a document is useless if the software application cannot communicate that information to a screen reader.

- Enhanced keyboard navigation

Most visually-impaired users will not be able to utilize the point-and-click interface that is so ubiquitous in software. Likewise, motor-impaired users need succinct mechanisms for interacting with applications. Applications must include the capability of accessing menus, commands and program functions through keyboard shortcuts and keystroke combinations.

- Ability to reformat, or reflow, document content

Most of us are so accustomed to using large monitors to view Web pages, documents, and software applications that it never occurs to us what would happen if our view into the world was suddenly reduced to a three-inch window. Yet, this is exactly the challenge that users with low visual acuity face. Magnifying the font size is not a viable solution since the viewer must scroll widely in all directions to cover the document. Instead, an application should reflow the magnified text into the available screen space such that the text reformats itself to conform with that space. Text reflow is also used, for example, to reformat a document for display on a Palm Pilot, although that is not a primary concern for accessibility. Text reflow is a good example of a feature that benefits users who do not have disabilities. Text reflow and logical read order are related to accessibility but they are handled separately. This fact will be illustrated with examples later in the course.

Instructor Notes: Point out that logical read order and text reflow are two different issues. Logical read order is, or can be, auditory, while text reflow is a visual phenomenon. Most of the course will concern logical read order. Text reflow will be covered in the last module.



How Acrobat Helps With Accessibility

We mentioned before that an accessible document must be able to give a visually-impaired person cues about the reading order of the text and alternate explanations of graphics and possibly tables. Adobe Tagged Portable Document Format (PDF) provides these capabilities.

Converting documents to Tagged PDF is a robust, long-term solution for creating accessible documents. Tagged PDF represents various components of a document, such as chapters, heading styles, blocks of text, tables, graphics, and so on, as tag elements. The tag structure is similar to markup languages such as HTML and XML. A document's structure is represented as a hierarchy of tag elements. The order in the hierarchy represents the reading order of the document. Since the content is represented with tag elements, other applications can extract the information and reuse it for other purposes. For additional technical information about Tagged PDF, please refer to section 9.7 Tagged PDF in the PDF Reference, Third Edition, version 1.4.

Instructor Notes: PDF Tags are essential for accessibility because they provide the hooks into the document that other applications, such as screen readers, can use. However, tags are only part of the total equation in the sense that the application must decide how to make use of the tag information provided to it. The document developer has no direct control over this part of the equation; screen readers and other accessible devices evolve and become more sophisticated over time. The primary goal of the document developer in terms of accessibility is to make sure that the available hooks are in place so that applications can access them for information about the document. PDF Tags generally provide more document information than is currently used by alternate access devices. Tags placed in a document today will be utilized in increasingly sophisticated ways as access technology evolves over time.

Tagged PDF offers the following benefits

- You can associate additional information with a particular tag element, such as a graphic, by filling in its alternate text property.
- Other applications can automatically reflow text and associated graphics to fit a page of a different size than was assumed for the original layout.
- The document's content can be converted to other common file formats (such as RTF, HTML, and XML) while preserving the structure and basic style information.
- It retains one of the primary advantages of PDF, the ability to preserve the exact look and feel of any source document, including all of the fonts, formatting, colors, and graphics, regardless of the application and platform used to create it.

Instructor Notes: The core of PDF tag support is that the logical order of the content of the document can be determined independently of the document's visual appearance and layout by traversing the tag structure hierarchy and presenting the contents of each node, which is what devices like screen readers do.

Tagged PDF documents are readily available to the general public since there are more than four hundred million copies of the free Adobe Acrobat Reader in use. Over two hundred government agencies worldwide have adopted PDF. Adobe Acrobat 5.0 improves the accessibility of functionality in the Adobe Acrobat software and the information in Adobe Portable Document Format (PDF) files.

Acrobat assists a user who wishes to read an accessible document by providing the following features:

- Provides usability enhancements, including enhanced keyboard shortcuts, support for high-contrast viewing, and the ability to zoom in and reflow text on the screen.
- Supports screen readers: Provides direct support for screen readers via the Microsoft Active Accessibility (MSAA) application programming interface (API) for Windows.® MSAA enables Acrobat 5.0 to integrate with assistive technology products including the newest versions of screen readers from vendors such as Freedom Scientific (<http://www.freedomscientific.com>), GW Micro (<http://www.gwmicro.com>), and Dolphin Oceanic (<http://www.dolphinoceanic.com>).

For users who wish to create accessible documents, Acrobat provides the features described below to both create and manipulate PDF files for accessibility:

- Optimizes legacy Adobe PDF files for accessibility: Transforms an untagged Adobe PDF file into a Tagged Adobe PDF file via the Make Accessible plug-in for Acrobat 5.0 for Windows (<http://www.adobe.com/support/downloads/detail.jsp?hexID=88de>). Resulting files can be read more clearly with assistive technology.
- Creates Tagged Adobe PDF files when converting Microsoft Office 2000 files, MS Office XP files, Adobe InDesign 2.0, Adobe FrameMaker 7.0, and so on to Adobe PDF, making it easier for people who use screen reader software to navigate a document in the proper reading order.
- Includes tools to help troubleshoot and optimize Adobe PDF files for accessibility, including the accessibility checker, the Tags palette, and more. The Tags palette lets an author review and touch up document reading order and add new tag elements to the document structure, such as form fields.
- Repurposes text: Supports export of text from a Tagged Adobe PDF file to Rich Text Format (RTF), or to XML, HTML, or TXT via a plug-in for Macintosh and Windows. Resulting files can be reused with a variety of applications including word processors and non-MSAA-compliant screen readers.
- Secures files while retaining accessibility: Provides new levels of security that enable an author to prevent copying and pasting of document content while still making it accessible to assistive technologies via MSAA.

In terms of the accessibility requirements for software applications, mentioned in the previous section, Adobe Reader 5.0 and Acrobat 5.0 provide the following forms of support:

- Keyboard navigation

Reader 5.0 provides users with keyboard shortcuts and key combinations for access to all program functions. In Acrobat 5.05, functionality that is also found in Reader is fully accessible. However, some functions unique to Acrobat 5.05, such as the freehand drawing tools used for electronic markup, are not available via keyboard.

- Logical order

The sequential order of the PDF Tags structure determines the order in which the elements of the document, headings, paragraphs, alternate text for graphics, and so on, are processed by devices like screen readers.

At the present time, only some of the information present in the Tags structure is available to screen readers. Consequently, screen readers primarily use the sequential positioning of elements in the PDF Tags structure. In the future, one would expect such devices, in conjunction with improvements in authoring software, to make use of additional structural information about the document, such as table heading tags and lists, to provide additional context about the document to visually-impaired users.

For that reason we encourage adding formal structure to electronic documents — selecting paragraph styles rather than formatting text with tabs, spaces, and carriage returns, for example.

- Content reflow

Acrobat 5.0 provides a TouchUp Order Tool for specifying the flow order of elements of the document when it is repositioned into a different layout or format. In regard to accessibility issues, the ability to reflow and reformat content, in addition to determining the sequential order in which that content is presented, allows a user with partially impaired eyesight to magnify the text in the document sufficiently to read it, and also to have it properly formatted within the viewing window and presented in the correct logical sequence.



Accessibility Design Considerations

Whether you're publishing a document in HTML, XML, Adobe PDF, or some other format, creating accessible documents requires much more than simply representing the original document accurately. Sighted people can look at a printed page and easily discern the difference between titles, subtitles, columns of text, headers, footers, and so on. Visual clues, such as location of the text on the page, bold text, and large font sizes help them determine the structure of a document so they can read and navigate it easily.

Unfortunately, assistive technologies such as screen readers cannot depend on these visual clues. They must instead rely on the underlying computer-based information to provide that same structure. As a result, making documents accessible depends on two things:

- Authoring the original documents so that they contain not just content, such as the text in the document, but also information about the structure of the content, such as how the text flows within the page and from page to page.
- Using publishing tools that can retain and encode both the content and the structure so that it can be interpreted by assistive technology.

In order to do this:

- Authors need to be aware of the importance of writing with the intent of creating accessible documents, and how to accomplish that.
- Authors need to be aware of the features within their authoring applications that support accessibility and make full use of them.

[Instructor Notes: Recurring theme: the advantage of controlling the PDF Tag structure from the original document using the original authoring software.](#)

We strongly recommend that you design for accessibility as much as possible in your documents and select publishing tools that support accessibility features. While there are tools that allow you to modify a Tagged PDF structure, they are currently time-consuming to use. Fixing a document's Tagged PDF can take anywhere from half an hour to an entire day, depending on the magnitude of accessibility problems that are found. In addition, it is not possible to save and reapply your fixes to a document's Tagged PDF on the next revision of the document. For example, you put in 4 hours of work fixing the Tagged PDF structure of a document. Three months later, your department finishes an update to that same document. You will have to redo the fixes on the revised document. Your goal is to minimize the amount of post-PDF processing that you need to do on each document.

[Instructor Notes: Software development in the area of structuring and tagging documents for repurposing and accessibility is probably at the first generation stage and will continue to evolve rapidly. If accessibility is a major concern in your documentation efforts, we suggest always using the latest versions of both your authoring software and Acrobat.](#)

We also recommend that you use the most recent versions of your authoring software. Older software does not do a good job creating Tagged PDF. Often, the resulting Tagged PDF requires a significant amount of work to fix problems that occur during the conversion process.

Document Design

In terms of actual document design, the most problematic areas are complex layouts and visual presentation of information. Visual presentation of information encompasses tables, pie charts, flowcharts, bar charts, graphics, pictures, and so on. If you intend on using a multi-columned layout, make sure that your authoring software has built-in tools to support this and that the content converts accurately to Tagged PDF. Layouts found in newspapers and magazines, where multiple unrelated articles appear on a page and continue on different pages, pose particular challenges. The authoring software must provide a mechanism to allow you to indicate the read order. In other words, you must be able to define where the continuation of each story can be found such that it is converted to Tagged PDF. Without this capability, you are faced with a very time-consuming task of defining the read order in the Tagged PDF so that it is accessible.

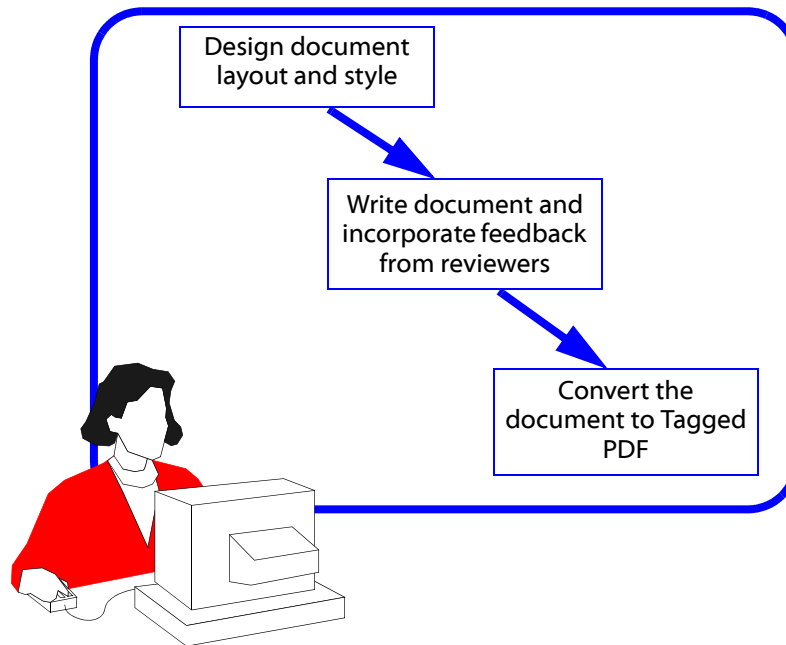
You should seriously evaluate the information that a visual is conveying. Is the information already present in the text? Is the visual simply providing color and images that are not essential to the message conveyed by the document? The alternate text that you tie to a visual should not repeat the caption. If the visual is a table, bar chart, pie chart, and so on, you should summarize the findings. For example, if a bar chart is showing the projected income over the next four quarters, state that and include the actual income figures in the alternate text. If a table is summarizing information, such as product requirements on different operating system platforms, you may want to consider including it in the text of your document.

Because accessibility is such a new area, the available authoring software varies in its capability to produce optimal Tagged PDF. For example, the work required to retain the reading order for a visually complex document, such as a brochure, may vary from application to application. Unless the logical structure of such documents is carefully prepared, usually involving iterative testing, it is not unusual for the sequence of the Tagged PDF structure to be almost but not quite right in terms of conveying the logical flow of the document's contents. You are always better off adjusting the original document to produce the desired PDF Tag sequence. Otherwise, you will have to adjust the Tagged PDF structure using the Acrobat Tags Palette.

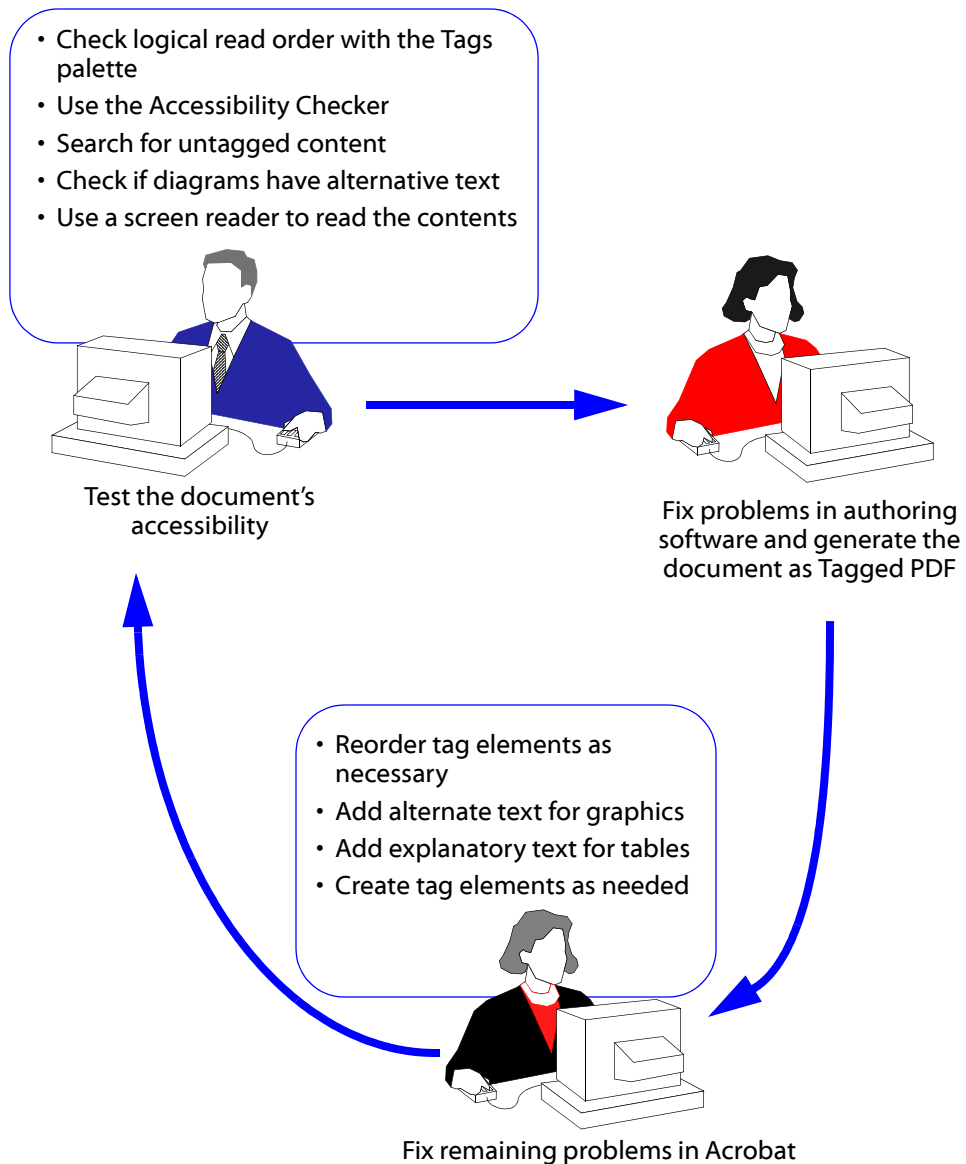
Workflow for Creating Accessible Documents

Creating an accessible document is an iterative process. Accessibility is still evolving. Expect that the techniques and guidelines will change over time as well as the capability of various software applications to support this process. However, the authoring software that you use as well as the types of documents your department produces will influence the workflow.

Within the authoring application, the user creates a document template that defines the overall look and appearance of the document. The template also includes the paragraph and character styles that will be used by the author. Once the document has been completed and reviewed, the author can generate a Tagged PDF file.



Once the author has created the Tagged PDF file, someone must verify that the content is actually accessible. The diagram below illustrates the workflow needed to fine tune a document's accessibility. Use Acrobat's Tags palette or save the file as **Text (Accessible)** to check the reading order. The Tags palette also includes other tools that allow you to search for page content that is causing problems, such as form fields and cross-reference links. You can use a screen reader to read the document the same way a visually-impaired person would access the document. Once accessibility problems have been identified, the author should return to the authoring application to fix as many of the problems as possible. Any remaining problems will have to be fixed using tools in Acrobat. For information on how to check a document's accessibility and fix problems within the authoring tool, refer to the next section, Module 2, in this course. For information on how to fix accessibility problems using Acrobat's tools, refer to [Module 3: Using the Tags Palette](#).



Before you release a document template for general distribution, test it with several existing documents that use a variety of different ways of presenting information. Work out any accessibility-related problems that you identify. This will eliminate the work that you will need to do post-processing the tagged PDF. There may be some issues that you identify that will have to be addressed by using alternate features within the authoring tool or by changes in people's work habits.

Module 2: Accessible Documents in FrameMaker 7



Introduction

This module provides an introduction and overview to producing accessible PDF documents with FrameMaker 7. You will be introduced to features of FrameMaker 7, and will acquire hands-on experience using it to create Tagged PDF. This module focuses on creating accessible documents for visually impaired persons. For additional information, you can also refer to the Web sites and other resources listed below in the [References](#) section for further material and samples.

Learning Objectives

At the end of this module, you will:

- Be familiar with the basic guidelines for using FrameMaker 7 to create accessible PDF documents.
- Understand how to configure the Acrobat Distiller for FrameMaker to produce Tagged PDF files.
- Know how to troubleshoot accessibility problems using tools available in Adobe Acrobat
- Be able to fix accessibility problems in a FrameMaker 7 source document
- Know how to add Alternative Text for graphics and other non-textual content elements, such as complex charts.
- Know how to use hidden alternate text to provide navigational or textual information for visually impaired readers.
- Be able to use available tools and methods to test the results of iterative changes to the original document.

Background Assumptions

Students who are taking this module should have a working knowledge of FrameMaker, although not necessarily FrameMaker 7, and should be familiar with using paragraph and character formats, anchored frames, and cross references.

References

General:

- Fully accessible web site: <http://access.adobe.com>
- White paper: *Accessible Design Guidelines*, Sarah Horton, Dartmouth College, May 2, 2002. Good treatment of many aspects of designing accessible documents: not specific to product. Many useful resource links.
<http://www.dartmouth.edu/~webteach/resource/download.html>

Product Information:

- Web site: *Recent FrameMaker Tutorials*
<http://www.adobe.com/products/tips/framemaker.html>
- Booklet: *How To Create Accessible Adobe PDF Files*
 - PDF Version:
<http://www.adobe.com/products/acrobat/pdfs/accessbooklet.pdf>
 - HTML Version: <http://access.adobe.com/booklet1.html>
- Booklet: *Advanced Techniques for Creating Accessible Adobe PDF Files*
<http://www.adobe.com/products/acrobat/pdfs/CreateAccessibleAdvanced.pdf>
- Booklet: *Adobe Acrobat 5.0 and Section 508*
http://www.adobe.com/products/acrobat/pdfs/Sect_5085.pdf
- Web site: *Acrobat Solutions for Accessibility*. Information about Adobe products specific to accessibility issues.
<http://www.adobe.com/products/acrobat/solutionsacc.html>
- Book: *Adobe FrameMaker 6.0 Classroom in a Book*. Adobe Press, (Peachpit Press, Berkeley, CA), 2000.

Contents

Topics	Exercises
Overview	—
Analyzing Documents for Accessibility	Generating Tagged PDF with FrameMaker 7
Assessing the PDF Tag Structure	Working with the PDF Tags Palette
Accessibility Testing	Testing Accessibility in PDF Documents
Making Graphics Accessible	Adding Alternate Text to Graphics Testing Alternate Text for Graphics
Working with Hypertext Links and Cross References	Using Cross References Using Hidden Cross References for Navigation

File to Download for Exercises in this Module

FMAccessibility.zip contains the following files:

- `sample_fm_doc_1.fm`
- `TermsOfUse.fm`
- `TermsOfUse.pdf`
- `PrivacyPolicy.fm`
- `PrivacyPolicy.pdf`



Overview

FrameMaker 7 is an extremely versatile document production and authoring tool that provides both Structured and Non-Structured (standard FrameMaker) modes of operation.

Standard (Non-structured) FrameMaker 7

Standard non-structured FrameMaker 7 can produce well-formed Tagged PDF documents, as well as FrameMaker (.fm) documents that are also accessible by devices like screen readers. Standard FrameMaker 7 can also save documents in HTML format, with cross references and hyperlinks intact, all of which are accessible by screen readers.

Structured FrameMaker 7

Structured FrameMaker 7 adds a new level of versatility to the document production process by generating XML representations of FrameMaker documents. Structured FrameMaker 7 is an update and enhancement to FrameMaker+SGML, and is, in effect, FrameMaker+XML. XML document representation is a common standard that is used to translate both contents and document structure between a number of output formats, such as HTML, XHTML, WML, and VoiceXML, with document formatting provided through the use of CSS and XSL-FO styling languages. One of the additional benefits of Structured FrameMaker is that it facilitates "round tripping" of documents during production, in that a document can be edited and modified in various supported formats, then translated back to an XML representation and opened in the original authoring tool with modifications intact.

Focus of This Module

This module focusses on using non-structured FrameMaker 7 to create accessible electronic documents in Acrobat PDF format. PDF provides a flexible platform for delivery of electronic documents using a wide variety of media. With the advent of Tagged PDF in Acrobat 5, alternative devices, such as screen readers and braille readers, can readily access PDF documents. Text within PDF documents can be correctly reformatted and reflowed when magnified for sight impaired readers. The ultimate goal in creating accessible PDF documents is to use the authoring tool, in this case FrameMaker 7, to produce a PDF tag hierarchy that reflects the logical read order and flow of the document. The PDF tag structure includes elements containing the text of the document, but it also must include elements like cross references, hyperlinks, and alternate text for relevant graphics.

[Instructor Notes: You might want to emphasize the first sentence in the following paragraph. Some students may be confused by the references to Structured FrameMaker and XML, and others may be curious about what it is. The salient point is that Structured FrameMaker and XML are irrelevant to the object of this module, which is to produce accessible PDF documents using FrameMaker as the authoring tool. Any FrameMaker 6 and FrameMaker 5 users in the class will have no problem making the transition to FrameMaker 7 in the way we use it here.](#)

The XML-based flexibility of Structured FrameMaker is *not* required to generate well-formed PDF tag structure. Standard FrameMaker 7 is so tightly coupled with PDF through the Acrobat Distiller that simply following good standard practices for document formatting, and correctly using anchored frames for graphics, is all that is required in FrameMaker to generate a logically correct PDF tag structure and a fully accessible document.

Documents created in Structured FrameMaker mode will also generate a logically correct PDF hierarchy, whether or not they have an XML tag structure. However, the process of creating a Structured FrameMaker XML tag structure for a document is non-trivial, and as useful as the XML tag structure is for other purposes, it adds nothing to the quality of the Tagged PDF structure. Since there is little point to creating a structured FrameMaker document without creating an XML tag structure for it, this module will focus exclusively on non-structured FrameMaker 7 documents.

Logical Structure of the Course

The module combines presentation with hands-on observation and exercises. It begins with an overview of some of the issues involved in producing an accessible document. You will open a sample FrameMaker file and follow the steps to save it as a Tagged PDF file. You will observe the PDF tag structure in the resulting document and will be introduced to some of the methods of testing to see if the logical flow of the PDF document is correct.

The module next covers adding descriptive text to graphics in a document, which devices like screen readers can read, and controlling the placement of that alternative text in the logical flow through the use of anchored frames.

The module concludes with some supplemental accessibility topics: adding alternate text to hidden graphics for navigation purposes and handling cross references in headers and footers.

At the end of this module, you will have been introduced to accessibility issues that are specific to FrameMaker 7, and you will have had hands-on experience with the typical steps required to produce accessible FrameMaker documents.

Sample Files and Exercises

[Instructor Notes: The sample file is included in the archived file FMAccessibility.zip, located in the \exercises sub-folder. Use WinZip or the equivalent to uncompress it. Since students will be primarily working with one file throughout, sample_doc_1.fm, suggest that they make a copy of their work in progress at various points as a fall back for any unrecoverable errors.](#)

The sample FrameMaker 7 file, `sample_doc_1.fm`, illustrates various points presented in this module. You will use it to create tagged PDF documents, and will modify it during the exercises. The sample document was taken from another module associated with this course: *Introduction to Accessibility Issues*. It contains certain elements that sometimes require special processing for accessibility purposes, such as graphics and cross references.

Overview of the Exercises

Instructor Notes: This is not a technically complex module, and there are no difficult concepts involved. Much of the work of converting FrameMaker documents to accessible PDF involves the kind of basic FrameMaker formatting practices that should be followed anyway. The exercises lead students incrementally through the steps of what would be a typical workflow for creating accessible PDF with FrameMaker. Students start with a sample FrameMaker file and work with it throughout the module, changing it in various ways and testing the results when the file is saved as a PDF file.

1. Generating Tagged PDF with FrameMaker 7—In this exercise, you will make a first pass through the workflow cycle, considering some of the design issues involved in creating accessible documents, and becoming familiar with, or reviewing, options for formatting FrameMaker documents. You will then export a FrameMaker document to PDF format and view the resulting PDF tag structure in the Acrobat Tags Palette.
2. Working with the PDF Tags Palette—In this exercise, you will become familiar with the Acrobat PDF Tags Palette and learn how to examine the logical flow of the Tags structure.
3. Testing Accessibility in PDF Documents—This exercise introduces you to various methods for testing and verifying the logical flow of the PDF document.
4. Adding Alternate Text to Graphics—In this exercise, you will use FrameMaker Graphics Object Properties to add alternate descriptive text that screen readers can read to illustrations, icons, and other graphic objects.
5. Testing Alternate Text for Graphics—This exercise covers methods for verifying the presence and functioning of alternate text for graphics.
6. Using Cross References—This exercise introduces you to creating accessible cross references, and covers some special issues involving cross references in Master Pages.
7. Using Hidden Cross References for Navigation—This is a supplemental exercise involving the use of hidden cross references to provide contextual and navigational information that is available to screen readers.

Certain other subjects, such as working directly in the Acrobat Tags Palette (in other words, post-PDF processing), and controlling the order of text reflow, are covered in a separate module in this course, *Using the Tags Palette*.



Analyzing Documents for Accessibility

General Issues

The first step in accessibility analysis is to determine what the logical read order of a document should be. This is a relatively straightforward process with FrameMaker documents, which tend to be laid out in a regular column order, compared to more free form documents that are laid out irregularly or in disconnected flows, containing mixtures of graphics and text. The FrameMaker-to-PDF process very accurately replicates the structure and flow of the original FrameMaker document, making the initial analysis of the document's logical flow easy and, in some cases, not really necessary.

Nevertheless, it is worthwhile to be aware of some of the issues and challenges involved in making visually oriented documents accessible to visually impaired persons. If a document is simply structured, a reader generally proceeds in a straightforward fashion, reading title, headings, and text line by line down the page. If the document has a more complex layout, or contains visually interesting elements such as photos, graphics, and so on, which compete for the viewer's attention, or which are referenced in the text flow, the appropriate logical order can be less obvious.

Sighted persons can process elements like graphics, sidebars, and callout boxes very rapidly and out of sequence, visually referring back to them multiple times if necessary. Screen readers, on the other hand, process everything in a document in sequence, and relatively slowly in comparison to the speed of visual reading. The sequence of presentation, therefore, is of much greater importance in documents that are accessible to screen readers than it is for documents whose unifying context can be partially expressed in terms of a visual layout within which sighted readers can easily shift and return in terms of attentional focus. The basic task of accessibility analysis of documents is defining the sequence in which the text in a document will be accessed, including alternate text for graphics and visual elements.

Beyond the basics, there are technical considerations as to how various parts of the document will be processed by accessibility devices like screen readers. The following are some of the issues:

- What should the screen reader do with document elements like corporate logos and purely decorative graphics? Should they be called out? Described? Ignored?
- A screen reader will not process text in the document that is actually in graphic format.
- How can a screen reader adequately convey the meaning of a graphic, such as an instructional photograph? What about a complex chart? A work of art?
- Tables are examples of informational formats that fully sighted persons can process in a near simultaneous manner, but which sequential devices like screen readers process slowly and somewhat tediously. How can the

elements of a table, as well as the overall meaning of that table, best be presented to visually impaired readers?

- If the text in a document is laid out in multiple column format, will the screen reader be able to follow the intended flow of the text, or will it read straight across multiple columns before going on to the next line? This is not generally a problem with FrameMaker documents.
- There are some pieces of information in a typical document that are required, but in reality are almost universally ignored, such as trademark and copyright notices, reprint policies, corporate information boilerplate text, the newspaper masthead, and so on. Sighted people make use of visual cues, such as position in the document, font type, font size, rules and borders, and the like, to recognize these elements and skip over them unless they have specific motivation to do otherwise. How should such information, and the option to access such information, be presented to visually impaired users?

FrameMaker 7 Formatting Guidelines for Accessibility

Instructor Notes: It is assumed that students who are taking this module of the course have a working knowledge of FrameMaker and are familiar with using such things as paragraph and character formats, anchored frames, and cross references. Much of this section and the accompanying exercise is a review of using paragraph formats. Adjust the pace of review according to the general level of students' experience with and knowledge of FrameMaker.

In regard to creating Tagged PDF, the most important structure in a FrameMaker document is the paragraph and the paragraph format. The Acrobat Distiller for FrameMaker translates FrameMaker paragraphs one for one to PDF tags, and uses the names of FrameMaker paragraph formats as PDF tag names.

Paragraphs are the structural building blocks of FrameMaker documents, and a FrameMaker document cannot be created without paragraph formats. Most FrameMaker documents are created from templates with defined paragraph formats, or paragraph formats are imported to the document, but even a new FrameMaker document created without a template has a default Paragraph Catalog with a set of default paragraph formats. If a plain text document is imported to FrameMaker, each paragraph is assigned the default format of Body. Even if a FrameMaker document consists entirely of default Body paragraphs, the Distiller will generate <_Body> PDF tags, with the underscore prefix added, for each paragraph in the FrameMaker document, in the same sequential read order as the original document.

The best policy for ensuring that an accessible PDF document with the correct logical flow can be generated from a FrameMaker document is to follow good formatting practices based on the appropriate use of paragraph formats throughout the document.

In addition to Tagged PDF creation, the correct use of paragraph formats is also the best policy for facilitating the creation of HTML from standard FrameMaker, and for generating XML forms of the document in Structured FrameMaker. A

thorough discussion of recommended FrameMaker formatting practices is beyond the scope of this document, but the following practices should be avoided:

Do Not:

- Use tabs to accomplish effects such as simulating rows and columns in tables, or creating multiple columns on a page, rather than using the **Insert Table** option to create tables, and using the **Page Layout > Column Format** menu to create multiple column layouts. If FrameMaker provides a layout mechanism, such as the Table Designer, use it: structural information about that element in the document will then pass to the Tagged PDF structure during generation of the PDF file.
- Add disconnected pages to the document without autoconnecting them to the appropriate text flow, or otherwise interrupting what should be a continuous text flow.
- Use blank lines, usually empty Body paragraphs, for vertical spacing between paragraphs rather than specifying Above Paragraph and Below Paragraph values in the Paragraph Designer. Using empty Body paragraphs for spacing does not create logical order problems in the PDF tag structure, but it unnecessarily complicates the tag structure because a PDF <_Body> tag is created for each blank line in the document.

Beyond paragraph formats, if you also use proper formatting practices to specify elements such as cross references, URL hyperlinks, bookmarks, and anchored frames, the attributes and characteristics of these elements are correctly transferred to the generated PDF tag structure.



Exercise: Generating Tagged PDF with FrameMaker 7

In order to do any of the exercises in this module, you must have Acrobat 5.x and FrameMaker 7.0 or higher installed on your machine, and you must have copied the sample files from **FMAccessibility.zip** to a folder on your machine.

Instructor Notes: This exercise reviews the basic notions of FrameMaker formatting, and introduces the process of saving a FrameMaker document as Tagged PDF. Students will not make modifications to the sample file in this exercise.

In this first exercise, you will take a look at a sample FrameMaker 7 document and save it as a Tagged PDF file. You will be introduced to the following:

- The format and structural characteristics of the sample FrameMaker document
- A way to begin looking at the document in terms of its logical flow
- Content objects in the document that need further processing for accessibility purposes
- How to generate Tagged PDF from FrameMaker 7

At the end of the exercise you should have an understanding of the following:

- The role of FrameMaker paragraph formats in generating Tagged PDF
- PDF tags and the structure of the PDF tag hierarchy

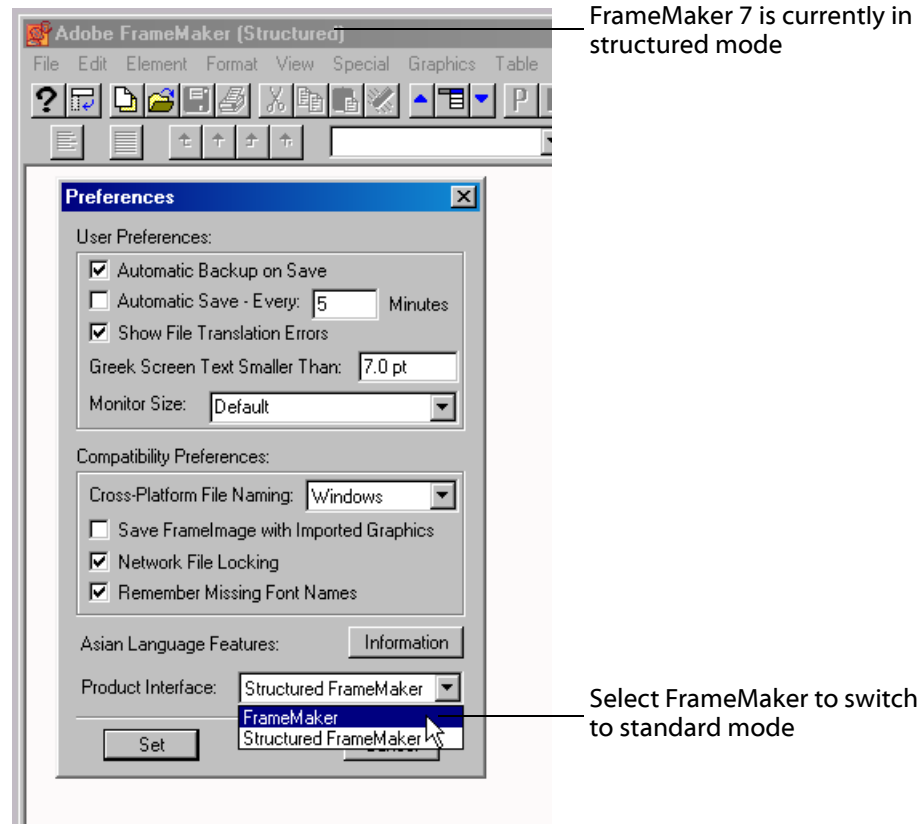
Opening and Inspecting the Sample File

FrameMaker 7 can be run in either of two modes: standard FrameMaker mode or Structured FrameMaker mode. The FrameMaker mode that you use has no consequence for the generated PDF tag structure. For the sake of consistency, start with and use the standard version of FrameMaker throughout the exercises in this module. Instructions follow:

1. Start FrameMaker 7 by clicking open the **Start** menu and selecting **Programs > Adobe > FrameMaker 7.0**.

If FrameMaker is in standard mode, the upper left title bar of the FrameMaker window displays the words *Adobe FrameMaker*. If it is in structured mode, the upper left title bar displays the words *Adobe FrameMaker (Structured)*.


2. If necessary, change the mode to standard FrameMaker:
 - a. Open the **File > Preferences > General** menu.
 - b. Click the **Product Interface** drop-down menu.
 - c. Select **FrameMaker**.
 - d. Click **Set**.

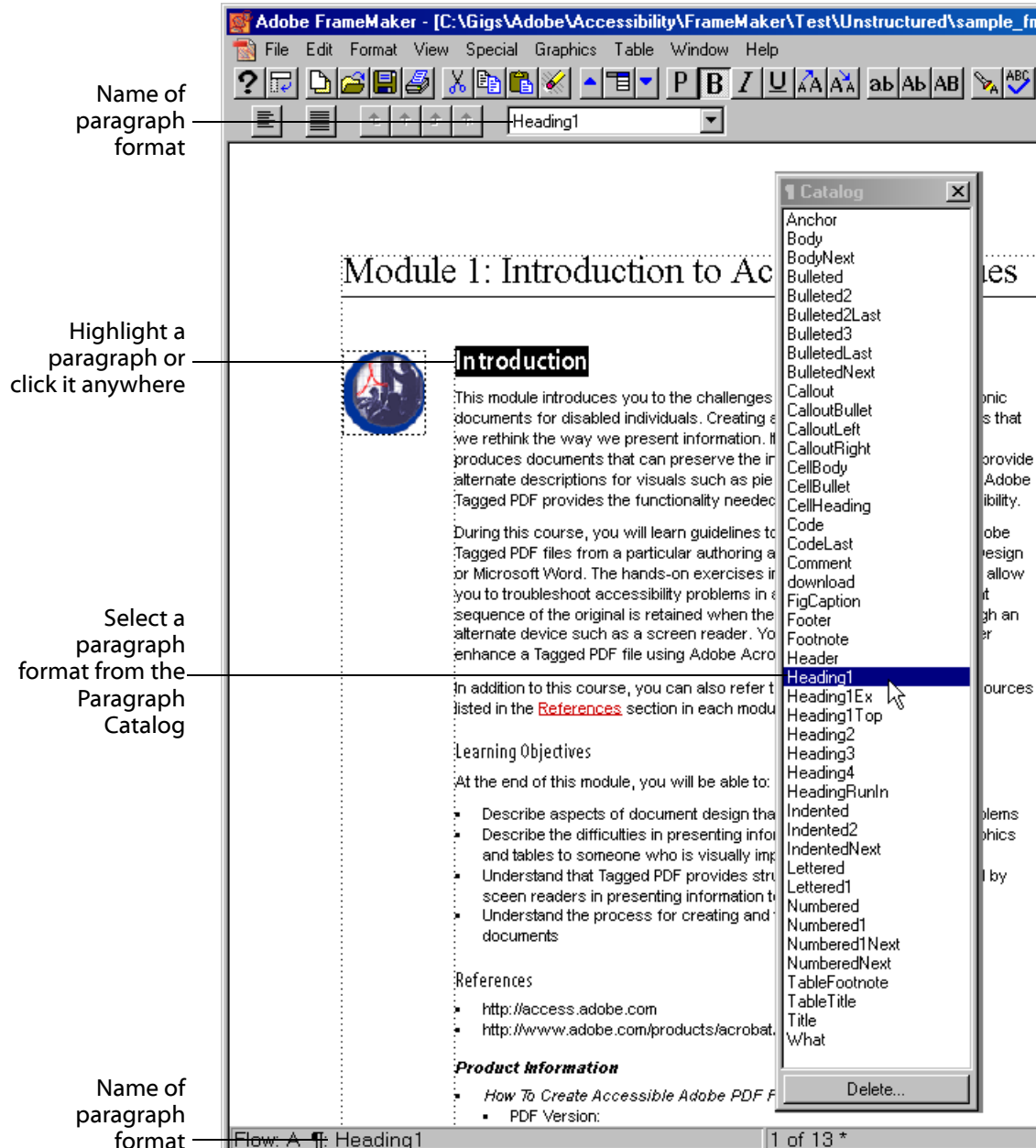


Note: If you change the existing mode from Structured FrameMaker to standard mode, you must shut down and restart FrameMaker 7.

3. Select **File > Open** and navigate to the folder containing the sample files.
4. Select **sample_fm_1.fm**.
5. Click **Open**.

The figure below shows the first page of the sample document is displayed in the screen shot below, as well as the Paragraph Catalog for the document.

6. Open the Paragraph Catalog by clicking the Paragraph Catalog icon () at the upper right border of the FrameMaker window or by selecting **Format > Paragraphs > Catalog**.



Review of FrameMaker Paragraph Formats

Most people who work with FrameMaker frequently or on a professional basis make regular use of paragraph formats to create formatting effects. You apply paragraph formats to paragraph blocks in the document through the Paragraph Catalog, which you opened in step 6. above. You can assign a paragraph format to text in the document by highlighting a paragraph, or by simply clicking anywhere in the text of the paragraph, and then selecting a paragraph format from the Paragraph Catalog.

The format you choose is applied to the selected paragraph. The name of the paragraph format appears in the drop-down menu field at the top of the FrameMaker window, and also in the status bar at the bottom left corner of the window. (Refer to the screen shot above). Clicking text anywhere in the document displays the name of its paragraph format in the drop-down menu field at the top of the window and in the status bar at the lower left.

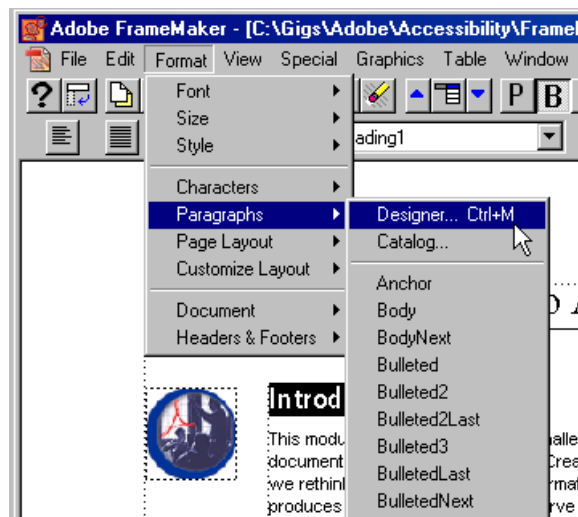
1. Take a minute to explore the paragraph structure of the document by clicking various sections and noting the paragraph format associated with each paragraph.

Note that formatting effects such as vertical spacing are accomplished by using special paragraph formats. For example, the Bulleted and BulletedLast paragraph formats are identical except that the BulletedLast format defines extra vertical space at bottom of the paragraph, which visually separates the last item in a bulleted list from the following paragraph.

2. Click the Introduction heading at the top of the document (refer to the figure above).

Note that it uses a Heading2 paragraph format.

3. Select **Format > Paragraphs > Designer**, or type **Ctrl+M**, to open the Paragraph Designer window for the Heading2 paragraph format.

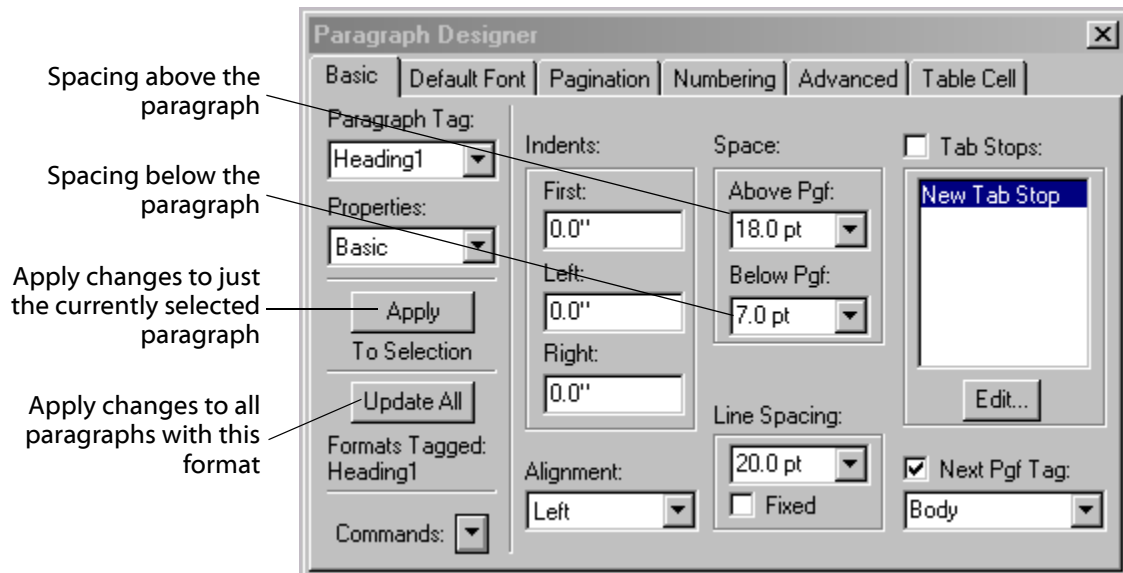


4. If you are not familiar with the Paragraph Designer, take a moment to note the formatting options available under the various Properties tabs.

You can completely control the visual characteristics and relative placement of text in a document the design and application of paragraph formats. For example:

- In the **Basic** tab you can specify the indentation and alignment of the paragraph, the spacing between multiple lines in paragraphs of that type, the tab stops for that paragraph format, and the vertical space to be placed above and below paragraphs of that type.

- You can specify the paragraph format to follow the current paragraph format when the Enter key is pressed.
- You can define characteristics such as font type and style, page placement, bullet and number formats, table formatting, and much more, from the various properties tabs in the Paragraph Designer.



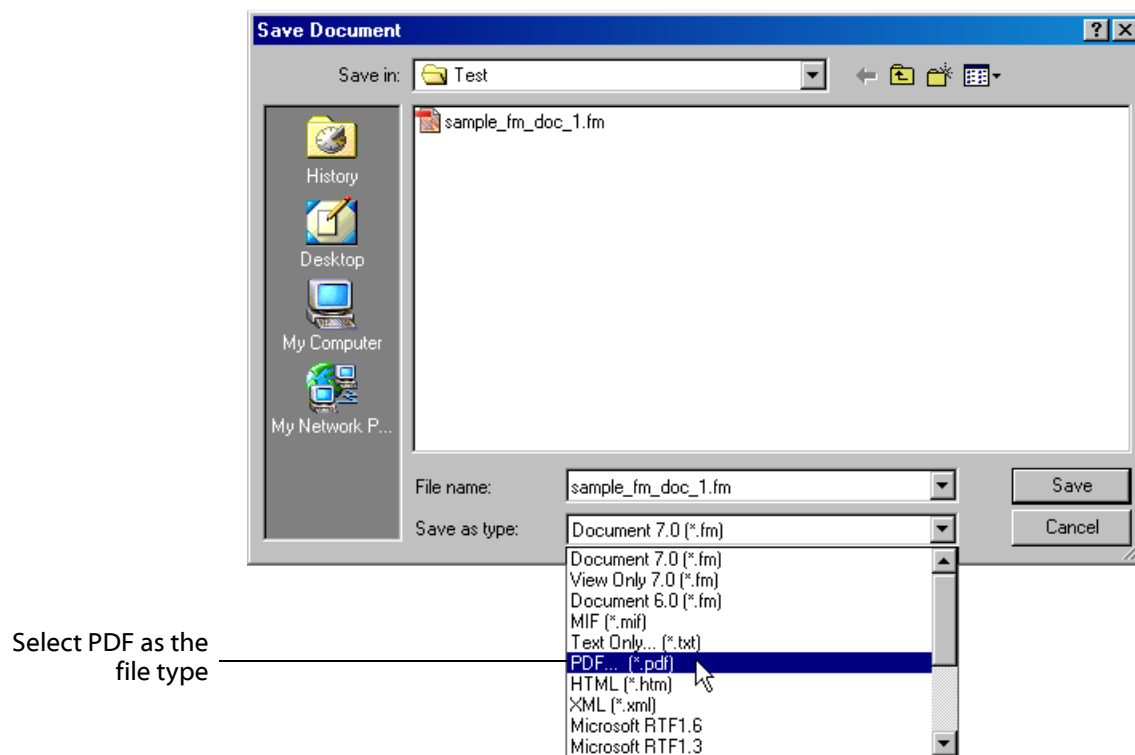
- To change existing properties of a paragraph format, enter or select the new properties, then click **Update All**.
 - You can override the paragraph format properties for a selected paragraph in the document by clicking **Apply**, but this should be avoided where possible for the sake of consistency.
 - Rather than overriding the **Below Pgf:** value for the last item in bulleted lists in order to create additional vertical space between the list and the next paragraph, use the **Commands > New Format** dialog in the Paragraph Designer to create a new paragraph format called, for example, **BulletedLast**, and then specify additional vertical space for the **BulletedLast** format in the **Below Pgf:** field.
5. Continue inspecting the sample document, keeping in mind some of the issues raised in the section on [Analyzing Documents for Accessibility](#).
- Note that the document contains cross references and hypertext links to URLs. You will need to make sure that these are functional and correctly positioned in the logical flow of the generated PDF tag structure.
 - The sample document also contains informative graphics, some with callouts and embedded text. You will need to provide alternate text descriptions for these graphics which convey the equivalent information to visually disabled users.
 - The document contains other graphics that call attention to various sections of the document, such as icons at the head of section topics and

exercises, and icons outside the left margin that call attention to special notes or cautions. You will need to decide whether to make these accessible to devices like screen readers, and if so, how.

- There are cross references in the Master page of the document that are outside the main text flow. These will probably be placed at the end of the PDF tag structure and may not be included in the logical flow at all. You will need to decide whether to include these links in the PDF tag structure and if so, how.

Saving FrameMaker 7 Documents as Tagged PDF

1. Select **File > Save As** to save the FrameMaker file with a name, location, and/or type that is different from the default `sample_fm_doc_1.fm`.



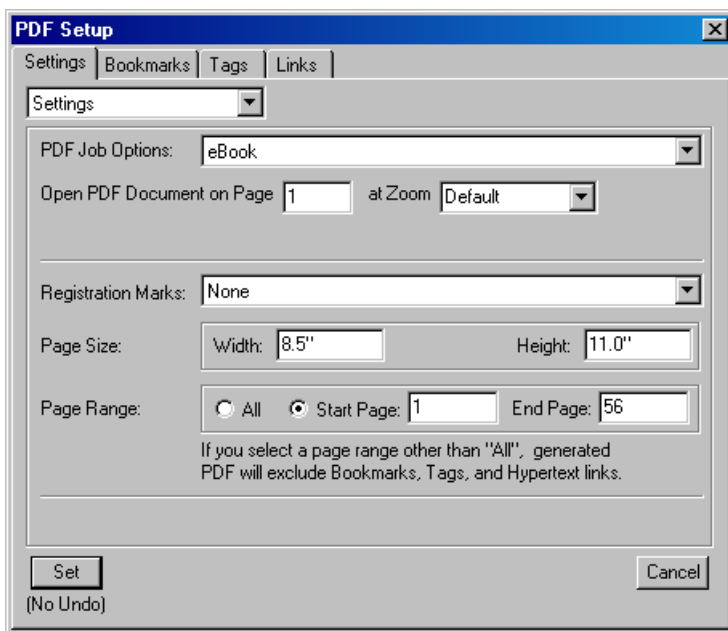
2. Make sure you select **PDF** as the **Save As** type.

A message box appears asking, in effect, whether you want to save the file as `sample_fm_doc_1.pdf`.

3. Click **Yes**.
4. Another message box may appear asking if you want to overwrite the existing `sample_fm_doc_1.pdf` file. If so, click **Yes**.

Saving a FrameMaker 7 file as a PDF file type invokes the Acrobat Distiller for FrameMaker. The PDF Setup dialog box appears, from which you can specify various PDF conversion options. For accessibility purposes, you will modify

only settings in the **Tags** tab in this exercise. You will leave the default PDF Job Options set to **eBook**, but be aware of the different settings and their uses:



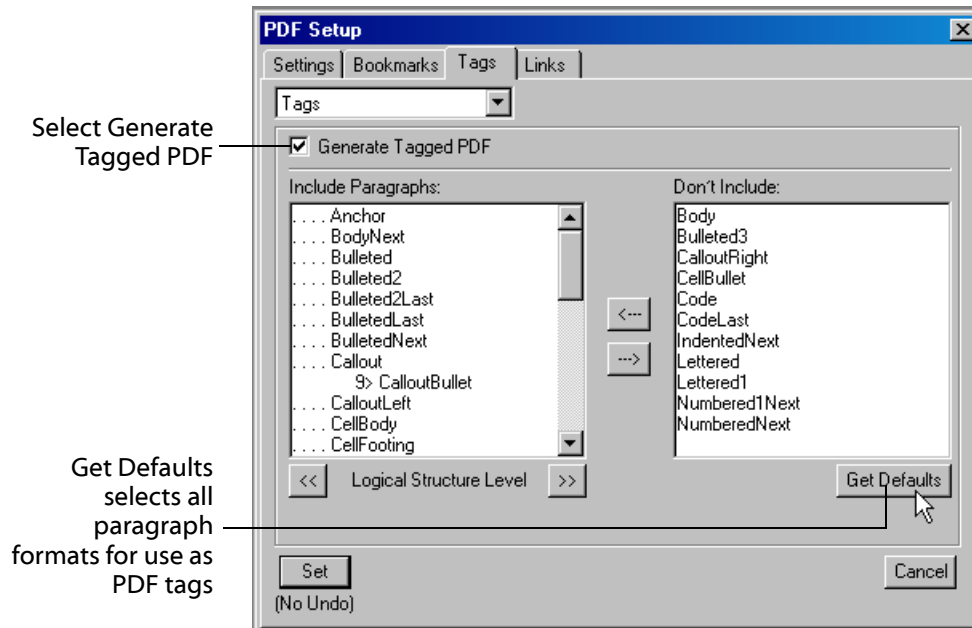
PDFMaker has the following sets of predefined job options:

- **eBook** job options are for Adobe PDF files that will be read primarily on-screen—on desktop or laptop computers or eBook Readers, for example. This is PDFMaker's default job option set.
- **Screen** job options are for Adobe PDF files that will be displayed on the World Wide Web or an intranet, or that will be distributed through an e-mail system for on-screen viewing.
- **Print** job options are for Adobe PDF files that are intended for desktop printers, digital copiers, publishing on a CD-ROM, or to send to a client as a publishing proof.
- **Press** job options are for Adobe PDF files that will be printed as high-quality final output to an imagesetter or platesetter, for example.

The **Screen** or **eBook** sets are generally more appropriate for documents during the review stage. For more information on setting and customizing PDFMaker job options, see the "Acrobat Distiller Options" chapter in the Adobe Acrobat online Help.

5. Leave the default selections in the **Bookmarks** tab as they appear.
6. Click on the **Tags** tab.
7. Select the check box **Generate Tagged PDF**.

You must do this to create an accessible PDF document.



The Include Paragraphs pane displays the names of paragraph formats that will be translated into PDF tags. The Don't Include pane contains names of paragraph formats in the current document that will not be translated into PDF tags. The arrow keys in the center permit selecting paragraph formats to include on an individual basis, while the **Get Defaults** button selects all non-selected paragraph formats.

8. Click the **Get Defaults** button at the lower right to select all non-selected paragraph formats.
9. Click the **Set** button at the lower left to start the PDF generation process.

You will examine the results in the next exercise ([Working with the PDF Tags Palette](#)).



Assessing the PDF Tag Structure

When you create a tagged PDF document, a tag hierarchy is also created that describes the document's structure. Tag elements are organized in a tree structure that reflects this structure. The PDF tag hierarchy starts with a Root node and expands into intermediate nodes that at some level correspond to FrameMaker paragraph formats. At the lowest level or end node of each structure tag is the actual content of the document itself, which could be text, a graphic reference, a cross reference, or other content element in the document.

The Acrobat Tags Palette

As you've already learned, the order of the PDF tag elements reflects the logical reading order of the document. The Acrobat Tags Palette provides the means for viewing and manipulating a document's Tagged PDF.

Once you have a tagged PDF document, you can use various Acrobat tools to verify the logical read order of your document and to make sure that the page content is truly accessible. Ideally, the final test of accessibility uses a screen reader. Some of the testing and verification methods you will be introduced to include the following:

- Stepping through a document using the Tags Palette to view the logical read order
- Using the Accessibility Checker to flag potential problems
- Searching for page content that is not included in the PDF tag structure
- Saving a document as Text (Accessible) to visually confirm the reading order
- Using a screen reader to check alternate text for graphics

You will learn more about how to use the Tags Palette to modify the PDF tag structure, and therefore the reading order, in the last module *Using the Tags Palette*. Keep in mind, however, that the object is to control the PDF tag structure from the original authoring tool, in this case FrameMaker 7, to eliminate or minimize the need to do post-PDF production modifications.



Exercise: Working with the PDF Tags Palette

In order to do any of the exercises in this module, you must have Acrobat 5.x and FrameMaker 7.0 or higher installed on your machine, and you must have copied the sample files from **FMAccessibility.zip** to a folder on your machine.

In this exercise, you will examine the tag structure that resulted from saving the sample FrameMaker file as a tagged PDF file. The exercise covers the following topics:

- How to view and interpret the Tags Palette in Acrobat 5.
- How to begin verifying the logical read order of the PDF document.

At the end of the exercise you should have an understanding of the following:

- What the PDF tag hierarchy looks like when viewed in the Acrobat Tags Palette
- How to use the PDF tag structure to assess the logical read order of the document

Viewing the PDF Tags Palette

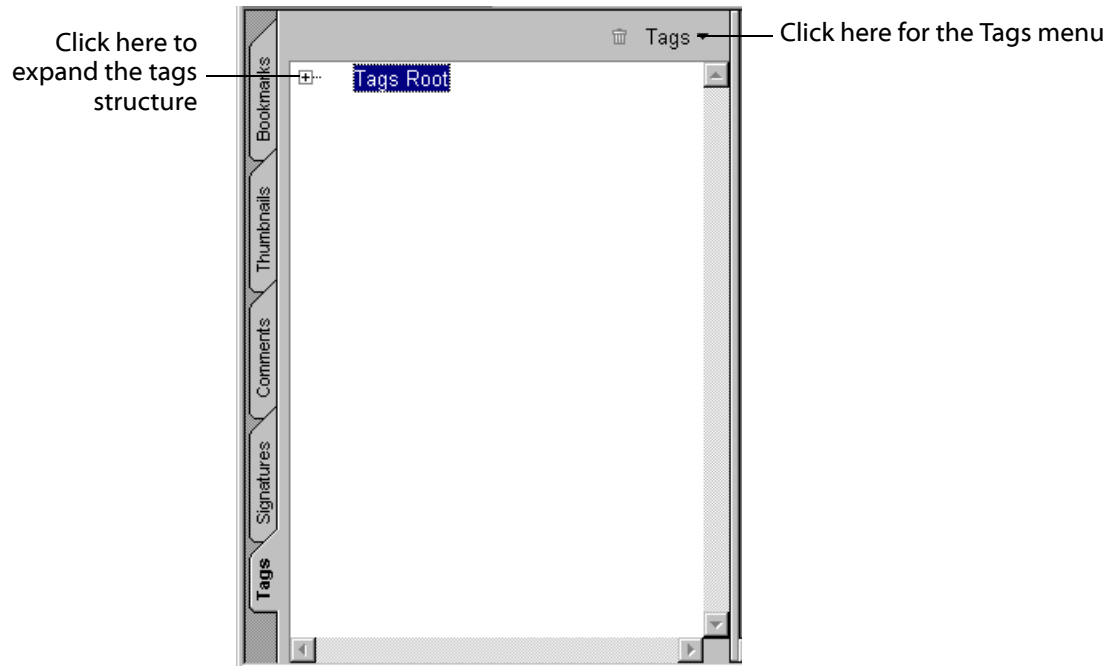
1. Open the PDF file, **sample_fm_doc_1.pdf**, that you generated in the last exercise using one of these methods:
 - Use **Start > Programs > Adobe Acrobat 5.0** to start Acrobat, then use **File > Open** from Acrobat and select **sample_fm_doc_1.pdf**.
 - Use Windows Explorer or My Computer to navigate to the folder containing **sample_fm_doc_1.pdf**, then double-click the filename to start Acrobat.
2. In Acrobat, select **Window > Tags** to open the Acrobat Tags Palette.



If this is the first time you have opened the Tags Palette, it may be floating detached from the main Acrobat window. If so, put the mouse cursor on the Tags tab on the Tags Palette, hold down the left mouse button, and move it into an empty space at the far left of the palette area in the main window in order to anchor the Tags tab there. You can then click the Tags tab to view the Tags Palette.

When you first open the Tags Palette for a tagged PDF document, you will see only the root node of the collapsed structure tree, as shown in the following figure.

There is a **Tags** drop-down menu at the upper right hand corner of the Tags Palette that contains a list of items you can use to find problems, or to modify the tag elements themselves or their placement. You will learn more about how to use the Tags Palette to modify the PDF tag structure, and therefore the reading order, in the last module *Using the PDF Tags Palette*.



3. Click on the + sign to expand the tree.
4. Right-click the Tags Root element in the Tags Palette and select **Turn On Associated Content Highlighting** from the menu.

As you select elements in the Tags Palette by clicking them, corresponding areas of the document highlight with a surrounding border.

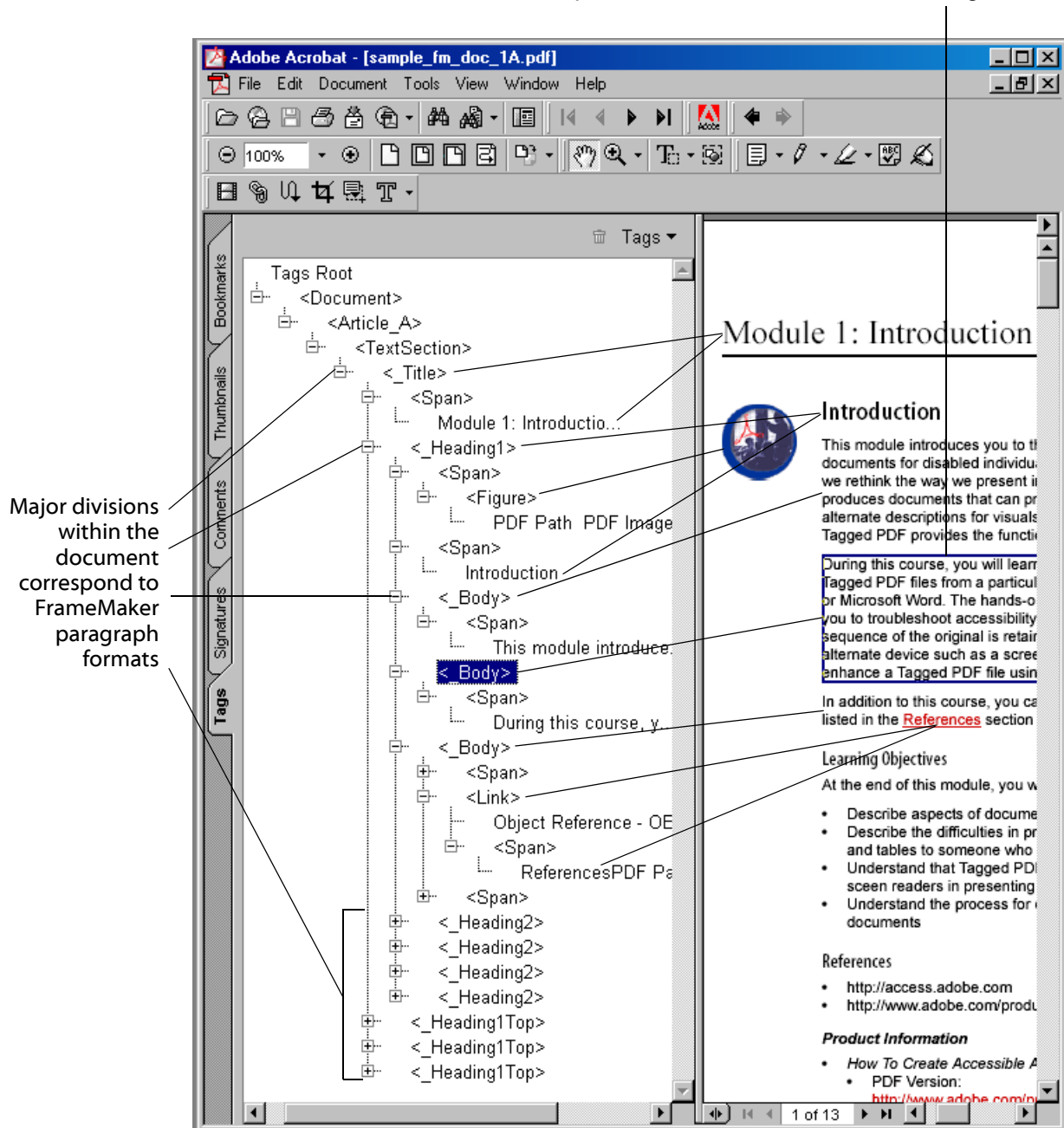
[Instructor Notes: Make sure students understand that step 5. describes a fundamental way of testing the logical read order of a Tagged PDF document.](#)

5. Continue expanding the PDF tag structure by clicking each tag's + sign, and note the sections of the document that correspond to each tag.

Work your way sequentially down the tag structure, keeping in mind that the sequential order of tags in the Tags Palette, and the document content they correspond to, determines the logical read order of the document for accessibility purposes. The only part of the tag structure that alternate access devices such as screen readers read (with the exception of alternate text for graphics, which you'll cover later) is the end node of each nested tag which contains a piece of the actual text content of the document.

Note that this operation—using associated content highlighting to see how the sequence of the PDF tag structure corresponds to sections of the document—is the first order of testing and validating the logical read order of the document.

Using Associated Content Highlighting highlights a section of text in the document that corresponds to the selected element in the Tags Palette



As you expand the PDF tag structure and note the corresponding sections of the document, you will see that the sequential order of the tag structure, for all practical purposes, corresponds exactly to the layout and expected reading order of the original FrameMaker document.

Certain default tags, such as **<Document>**, **<Article>**, **<TextSection>**, and ****, act as wrappers for other tags that correspond to paragraph formats

in the original FrameMaker document, and have the same names with an added underscore prefix.

Instructor Notes: Turning on Associated Content Highlighting disables all other options in the Tags Palette right-mouse button menu. To access the other options, students must turn Associated Content Highlighting off.

instructor Notes: Point out that the last node of each tag in the hierarchy is the actual content of the document. If the last node of a tag has no text content or object reference, then it generally points to a blank space and can generally be ignored.



Accessibility Testing

Instructor Notes: This is an introduction to testing methods. Emphasize that thorough testing requires the use of multiple testing methods. Point out that screen readers and saving as text are useful methods for checking logical flow, but that they won't provide information about untagged elements in the document. The Accessibility Checker will do the latter, but is not useful for testing logical flow.

There are several methods for testing the logical flow of the PDF document, each with strengths and shortcomings. The best approach when testing for accessibility compliance is to use more than one method.

Stepping through the Tags Palette

In the previous exercise, [Working with the PDF Tags Palette](#), you used one method of testing logical flow, that of verifying the reading order by clicking down the Acrobat Tags Palette with Associated Content Highlighting turned on, and checking the corresponding areas of the document.

Accessibility Checker

Acrobat provides a tool called the Accessibility Checker, which reports on possible accessibility problems in a document and marks areas of the document where those problems occur. The Accessibility Checker does not look for problems with the logical read order. It can, however, look for the following types of problems:

- No language specification

In the future, screen readers may be able to change languages from document to document. Specifying the language in which the document is written will provide the necessary information. Currently, assisting technologies do not utilize this feature.

- No alternate explanations for graphics

Visuals that convey important information for a document should have an alternate text explanation for visually-impaired people. There are some cases where you may choose not to provide alternate text because the graphic does not contribute any significant content, such as a company logo or border graphics for a page.

- Characters that do not have a corresponding Unicode encoding

Unicode provides a means to avoid problems caused by the existence of conflicting character codes found in the hundreds of separate character coding systems used throughout the world. In some cases, based on the way a font is encoded, there may be missing Unicodes for certain characters. The Accessibility Checker flags those characters because the text that corresponds to the character cannot be reproduced when the encoding is missing, and a therefore a screen reader cannot read it.

- Document components, like text, graphics, and so on, that have no corresponding tag element

There may be occasions when the conversion process is unable to convert text, symbols, or visuals into a corresponding tag element. You should investigate these errors since any content without an associated tag element will be invisible to assistive technologies such as screen readers.

- Form fields that have no descriptions

If your document is an electronic form, visually-impaired users need explanations for each of the data fields. Otherwise, it is not clear what kind of information they have to provide. This course does not cover specific issues pertinent to making electronic forms accessible.

While the Accessibility Checker can be a helpful tool, you need to be the final judge on whether a problem warrants fixing.

Saving the Document as an Accessible Text File

When you save a Tagged PDF document as a **Text (Accessible)** text file from Acrobat, the content is printed to the file sequentially, line by line, without formatting, in the same order as the tags in the Tags Palette. Therefore, the order of the text in the text file represents the logical order of the document.



Text (Accessible) is only available with Acrobat 5.0.5 or greater running under Windows. You can save the PDF file as Text (Plain), but Alternate Text for graphics may or may not be displayed. Actual Text is displayed in both Text (Accessible) and Text (Plain) files.

Screen Readers

Instructor Notes: The use of a screen reader is almost essential for complete and accurate testing of document accessibility. We strongly recommend installing at least one for use by the class. Ideally, each student workstation would have a screen reader installation along with a headset to reduce noise. Alternately, the instructor can have a single copy of the screen reader installed on a workstation and can demonstrate its use to the entire class at appropriate times. In the latter case, the instructor should call for students' attention toward the end of testing exercises, and use the screen reader to demonstrate the results of document modifications made during the exercise.

Screen readers can be difficult and confusing to use. We recommend that you spend some time practicing with one before delivering the course. Learn the basic navigation commands needed to use a screen reader for the kinds of testing described in these exercises, such as start at the top, start over, read next line, read last line, jump to hyperlinks, and so on. Free demo downloads of popular screen readers, with documentation, are available from the URLs listed below.

Using a screen reader to read a tagged PDF file is the benchmark of accessibility. Obviously, if you don't hear the content that you expect, something is wrong. While we do recommend that you use a screen reader to test your document, we

also recommend that you use it selectively or save its use for last. Listening to a screen reader process a document can be a time-consuming and tedious task. You will need to refer to the screen reader's documentation to understand how to use it. Expect to invest some practice time learning whichever screen reader you select.

Some major vendors of screen readers, GW Micro, Freedom Scientific, and Dolphin Oceanic among them, offer free downloadable demonstration versions of their products. You can use the demonstration version for 30 minutes, then you need to reboot your machine to use it again. You can try them out at

- GW Micro - <http://www.gwmicro.com/demo/>
- Freedom Scientific - http://www.freedomscientific.com/fs_downloads/jaws.asp
- Dolphin Oceanic - <http://www.dolphinusa.com/download/demos.htm>



Exercise: Testing Accessibility in PDF Documents

In order to do any of the exercises in this module, you must have Acrobat 5.x and FrameMaker 7.0 or higher installed on your machine, and you must have copied the sample files from **FMAccessibility.zip** to a folder on your machine.

In this exercise, you will be introduced to the following methods for assessing the logical flow and readability of the PDF document:

- The Acrobat Accessibility Checker
- Saving the PDF document As Text (Accessible)
- Using a screen reader

At the end of the exercise you should have an understanding of the following:

- The characteristics and strengths of various methods for assessing logical reading order

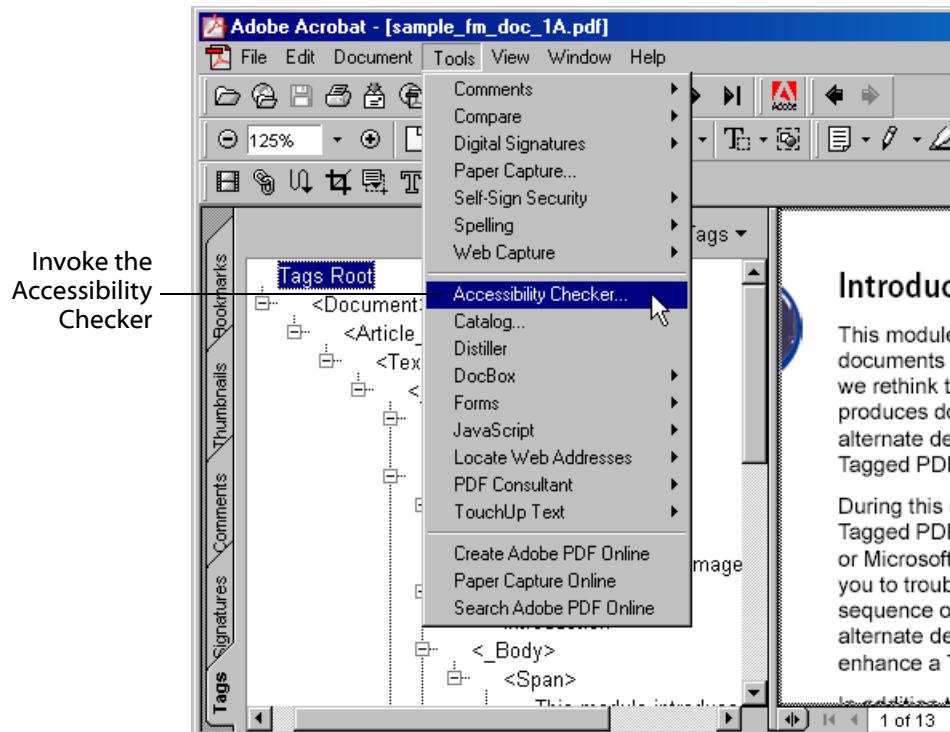
Using the Acrobat Accessibility Checker

The Acrobat Accessibility Checker is a tool that analyzes a document and identifies potential problems that could impact accessibility.

Instructor Notes: Mention that this will be an introduction to the Accessibility Checker and that there is more to it than we can cover in detail here.

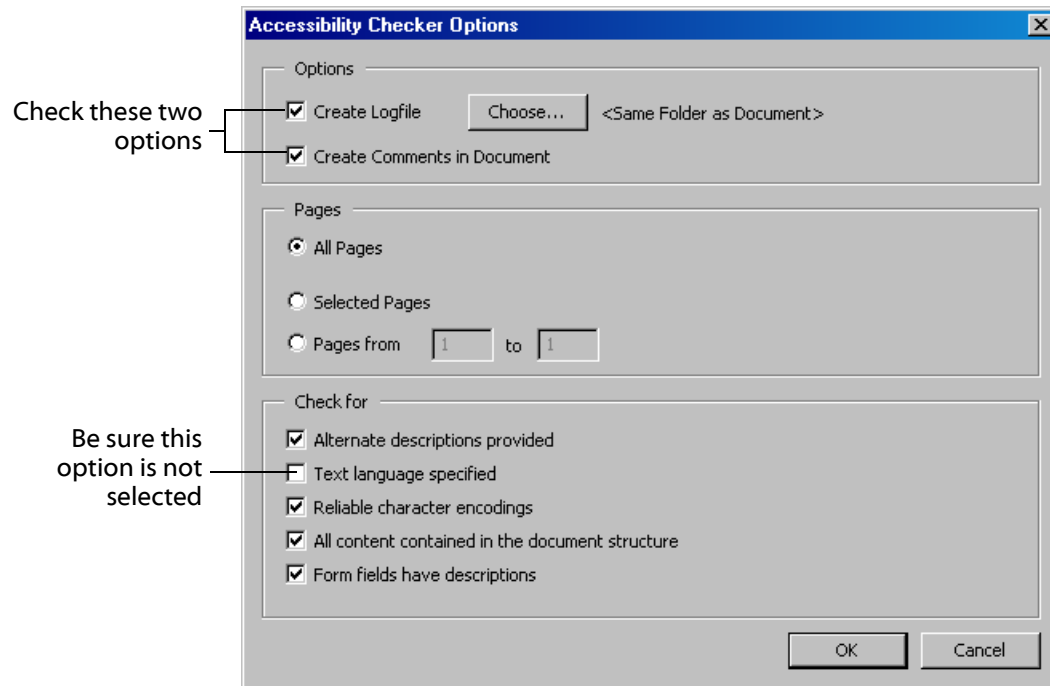
Instructor Notes: In some ways, the main point of the following exercise is to give students some sense of what kind of Accessibility Checker warnings are useful and which ones they can ignore. In this exercise, the Accessibility Checker will generate a slew of spurious warnings about graphics that lack alternative text. It assumes every line and curve in two drawings are separate graphic elements. However, the warnings are useful as a reminder that some of the flagged graphics in the document do need alternate text. The warning about the cross reference in the Footer being outside the text flow is also useful.

1. Start the Accessibility Checker in Acrobat by selecting **Tools > Accessibility Checker**.

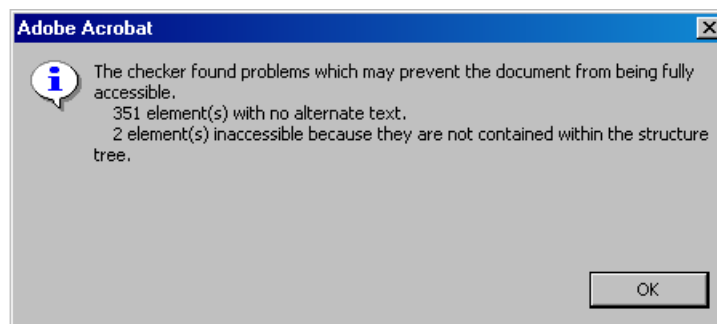


The Accessibility Checker Options dialog box appears. In this window, you can:

- Direct the Accessibility Checker to process particular pages, a range of page or the entire document.
 - Specify that Acrobat write out its summary into a log file.
 - Configure the types of problems that the Accessibility Checker searches for. By default, the Accessibility Checker reports a summary of its findings in a dialog box. While the summary reports the number and types of problems found, it is difficult to correlate those problems back to specific locations in the document. You should always enable comment creation, which allows the Accessibility Checker to highlight the offending document element and attach a comment describing the type of problem it found. You can easily view the problems using the Acrobat Comments Palette. For more information on how to use the Comments Palette, see the online help included with Acrobat 5.
2. Check **Create Logfile** to create a log file of accessibility warnings that you can later reference.
 3. Check **Create Comments in Document**.
 4. In the **Check for** section, be sure that the Text Language Specified toggle is unchecked, indicating that any language checking is turned off. The remaining toggles should be checked.



5. Click **OK** to start the Accessibility Checker process.



Although the initial report appears to be discouraging, having 351 elements with no alternative text, it is actually not as bad as it looks. This will turn out to be a case of the Accessibility Checker being a bit misleading by reporting on potential problems too thoroughly. Since you have turned on the comment creation, you can page through the document to find out where the reported problems are located.

6. Click the **Comments** tab at the left side of the window to display the Comments Palette.
7. By default, comments are displayed by page. If yours are not, click on arrow at the upper right of the Comments Palette to open the **Comments** drop-down menu and select **By Page**.
8. Click each page element in the Comments Palette to expand it.

Notice that as you click each comment, the rectangular frame around the corresponding area is highlighted in the document. The rectangular frames, usually red, mark elements in the document that the Accessibility Checker considers suspect for accessibility purposes.

Most of the warnings refer to graphics in the document that do not have alternate text descriptions:

- *When non-text elements do not have text equivalents, their content is lost to screen readers and environments with limited graphics capabilities.*

In other words, screen readers do not process graphics that have not been given alternate text descriptions. Since you have not yet added alternative text to any graphics, this should not be particularly surprising. You also know that there are some graphics in the document, but how can there be 351?

9. Expand the comments in the Comments Palette for Page 1 and click on one of the *When non-text element...* messages.

Notice that the icon outside the left margin next to the Heading is highlighted. The icon is a graphic, of course, but the question is, do you want to provide it with alternate text? This is a design decision. Usually, elements like the icon in this example are considered to be *artifacts* that are superfluous to the meaning of the document.

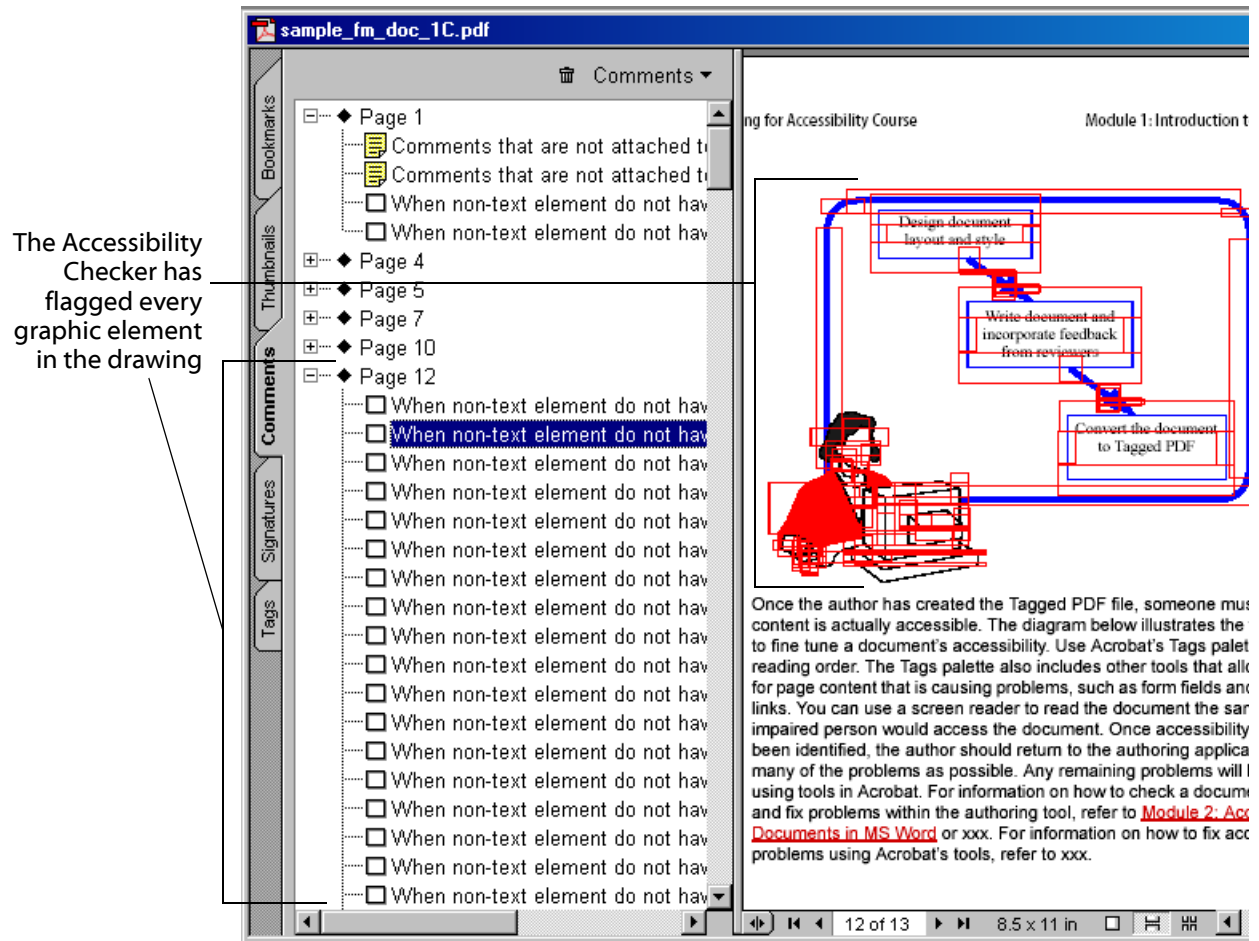
In general, you do not want to provide artifacts with alternate text. On the other hand, some visual artifacts provide context and navigational aids to sighted readers. In such cases, you may want to attempt to duplicate the functionality by, for example, providing alternate text that reads *Start of section*, or *End of article*. You will consider alternate text in more detail and decide how to deal with artifacts in the next exercise ([Adding Alternate Text to Graphics](#)). For now, mentally note that the Accessibility Checker flags every section heading and exercise heading icon as not having alternate text.

10. Expand the Page 12 element in the Comments window.

Notice that most of the 351 warnings are on pages 12 and 13.

11. Click one of the comments under Page 12.

Notice that many of the distinct elements that comprise the graphic are highlighted with red rectangular boxes. The Accessibility Checker treats every line, curve, arrow, and text box in the drawing as a separate graphic element and flags it for not having alternate text. This is definitely overkill. In the next exercise ([Adding Alternate Text to Graphics](#)), you will consider some alternatives for satisfying the Accessibility Checker's complaints, including doing nothing.

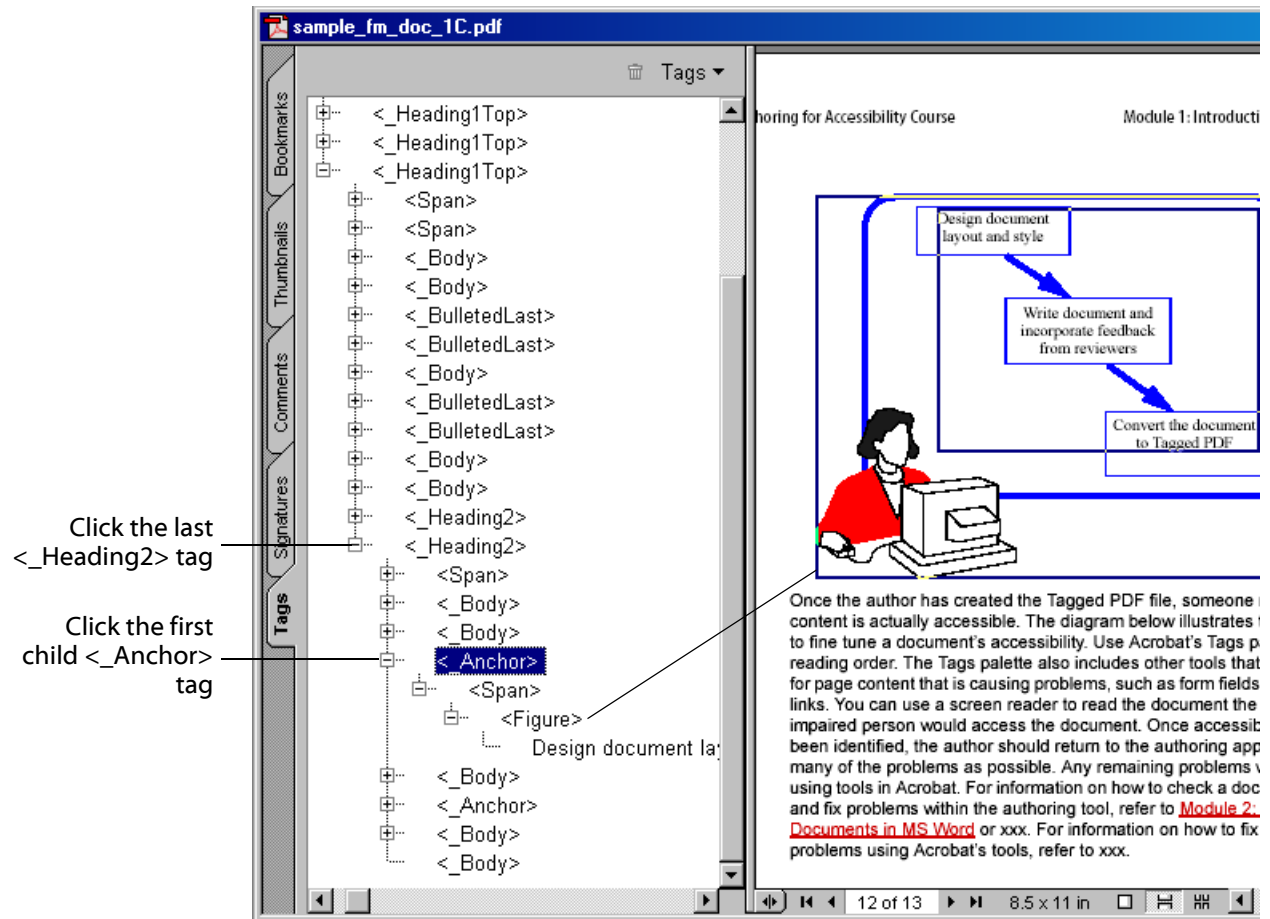


12. Click on the **Tags** tab in the Palette Window to display the PDF tag structure.
13. Select **Tools > Comments > Delete All** and click **OK** to remove the highlighting the Accessibility Checker has placed around suspect items in the document.
- This is just to simplify the display.
14. Expand the tags hierarchy to the Heading level.
15. Click the last **<_Heading2>** tag in the structure to expand it.
16. Click the first **<_Anchor>** tag to expand it.
17. Right-click a tag and make sure **Turn On Associated Content Highlighting** is selected.

Notice that the graphic in the document is represented in the PDF tag structure as a single **<Figure>** tag wrapped inside an **<_Anchor>** tag. (All PDF tags derived from paragraph formats have **** child elements.)

Although the Accessibility Checker sees 351 graphics in the document, Acrobat Distiller has actually handled the PDF tag translation from FrameMaker correctly, placing one and only one **<Figure>** tag inside an

<_Anchor> tag, which is the way the FrameMaker document was constructed.



The callout text inside the <_Figure> tag will not be a part of the logical read order. You must supply <_Figure> elements with alternate text before screen readers can access them.

So far, the Accessibility Checker has revealed to you some things you know, that certain graphics in the document should have alternative text, along with a number of things you did not need to know. There is one other matter, on Page 1, that should be investigated.

18. Use **Tools > Accessibility Checker**, with **Create Comments in Document** selected, to re-generate Checker Comments.
19. Click the **Comments** tab in the Palette window to display the comments.
20. Expand the Page 1 element.
21. Click one of the page icons that start out *Comments that are not attached...*

Notice that the comment points to a cross reference in the footer of the document.

22. Double-click the comment icon in the document to open a comment window containing the entire text.
23. Open the original FrameMaker sample document (`sample_fm_doc_1.fm`) in FrameMaker 7, with a view of page one and the cross references.
24. In FrameMaker, select **View > Master Pages**.

Notice that the footer text and the cross reference are really part of the Master Page layout—they are not elements that are connected with the main text flow, usually *Flow A*, in the FrameMaker document. The Accessibility Checker is telling you that these elements will not be accessible by a screen reader. That may or may not be the case, but the alert is useful. You will test it later using different methods. If the cross references are not included in the logical flow, you will need to decide what to do about them, if anything.

Accessibility Checker Summary

The Accessibility Checker alerted you to elements in the document that are outside the text flow and may not be included in the logical read order. It alerted you to many suspect graphic items that lack alternate text, although there are really only three figures that lend additional information to the content of the document, and which therefore should have alternative text, on pages 5, 12, and 13.

The Accessibility Checker occasionally issues false alarms. It will not help you to assess the logical flow of the document. The most successful strategy for evaluating accessibility and logical order in a PDF document is to use multiple methods; the Accessibility Checker is one such method. You will be introduced to other methods in the following exercises.

Saving the Document as an Accessible Text File

When you save a Tagged PDF document as an accessible text file from Acrobat, the content is printed to the file sequentially, line by line, without formatting, in the same order as the tags in the Tags Palette. Therefore, the order of the text in the text file represents the logical order of the document.

1. In Acrobat with the file `sample_fm_doc_1.pdf` open, select **File > Save As**.
2. Choose **Save as type: Text (Accessible)** and name the file `sample_fm_doc_1.txt`.
3. Click **Save**.
4. Open the file with a text editor and look at the sequence of text.

You will see that it corresponds in sequence to the expected reading order in the original document and in the PDF document. You can expect the FrameMaker Acrobat Distiller process to accurately translate an uninterrupted text flow into sequential tags, no matter how the text flow is visually formatted on the page. Therefore, you want to look for things that should be included in the Text (Accessible) file, but are not.

For example, if you search (**Edit > Find** or **Ctrl-F**) for the cross references contained in the Master Pages, *Terms of Use* or *Online Privacy Policy*, you will not find them. This almost certainly guarantees that the text and the URL links will not be accessible to devices like screen readers. You will consider the options for handling this in a later exercise.

Using a Screen Reader

In the same way that saving a document as accessible text is useful for revealing what is not included in the logical read order, but should be, using a screen reader serves the same purpose in testing a document's accessibility.

1. Depending on the training environment, start an installed screen reader (see [Screen Readers](#)), or have the instructor do so.
2. Use the screen reader's controls to navigate to the top of the PDF document.
3. Let the screen reader read through one or two paragraphs.

Try to imagine what it would be like trying to make sense of this document as a visually impaired person using a screen reader.

Further exploration of the screen reader will be postponed until the next exercise, when you will use it to test descriptive text applied to graphics in the document.



Making Graphics Accessible

As you saw in the last exercise with the Accessibility Checker, there are graphic elements in the document that need alternate text descriptions in order to be processed by devices like screen readers. Setting aside for a while the issue of icons at the beginning of exercises or sections, such as the one outside the margin to the left, there are three graphics in the sample document that lend additional information and meaning to the substance of the document.

- The first one, on page 5, illustrates the location of various elements in a document, such as Logos, the Title, Subtitles, Columns, Images, and Tables.
- The figure on page 12 illustrates the initial development process, from document design, through the writing process, to the creation of a Tagged PDF document.
- The figure on page 13 illustrates the typical iterative development and test process used to create a well-designed accessible document, from testing, through fixing accessibility problems in the original authoring software and re-testing, to fixing the PDF tag structure itself directly in Acrobat if necessary.

You will provide these three graphics with alternate descriptive text.

There are also many graphic elements in the document that were flagged by the Accessibility Checker for not having alternate text. You will consider the options for handling artifacts such as these, and also for handling elements like the icons that mark the beginning of sections and exercises.

Adding Alternate Text for Context and Navigation

Visual cues help sighted readers to orient themselves within the context of a document. In this document, for example, certain font types and font sizes are used as section headings, and these visual cues along with the actual content of the text, help sighted readers to understand that a shift in context and subject is about to take place. Likewise, the icons outside the left margin at the start of each topic section and exercise section in this document provide similar contextual cues to sighted readers. Although screen reader technology is advancing along with developments in document accessibility, you cannot count on a given screen reader to make use of such visual cues, for example, to read Headings differently than Body text, or to announce new sections in a document.

As document designer, you can decide whether a visual element in a document is simply an embellishment that adds no pertinent information to the document, or whether it serves a useful purpose in terms of adding information about the content of the document or about the structure of the document itself.

Alternate Text in FrameMaker 7

In FrameMaker 7, you can create alternate text for graphics by assigning an attribute-value pair to the anchored frame containing the graphic. Two forms of alternate text attributes are used for graphics.

- The Actual attribute is used for text rendered in graphic format, for example, a bitmapped title or logo. The Actual value should be the equivalent of the graphic text as it would be read.
- The Alternate attribute is a description, which can be detailed, of the graphic or its content.

Keep these points in mind:

- Actual attribute text should be short, 150 characters at the most, and ideally much shorter. It should be used only for graphically rendered text, and it should replicate that text exactly.
- Alternate attribute text can hypothetically be as long as necessary. It is used to provide annotations or detailed alternate text descriptions of graphics which convey relevant information in the document.
- Screen readers process both Actual text and Alternate text. It should never be necessary to use both, but if both are present, Alternate text takes precedence.



Exercise: Adding Alternate Text to Graphics

In order to do any of the exercises in this module, you must have Acrobat 5.x and FrameMaker 7.0 or higher installed on your machine, and you must have copied the sample files from **FMAccessibility.zip** to a folder on your machine.

In this exercise, you will be introduced to the following:

- The Alternate and ActualText attributes
- Adding alternate text attribute values to anchored frames for graphics
- Handling graphic artifacts

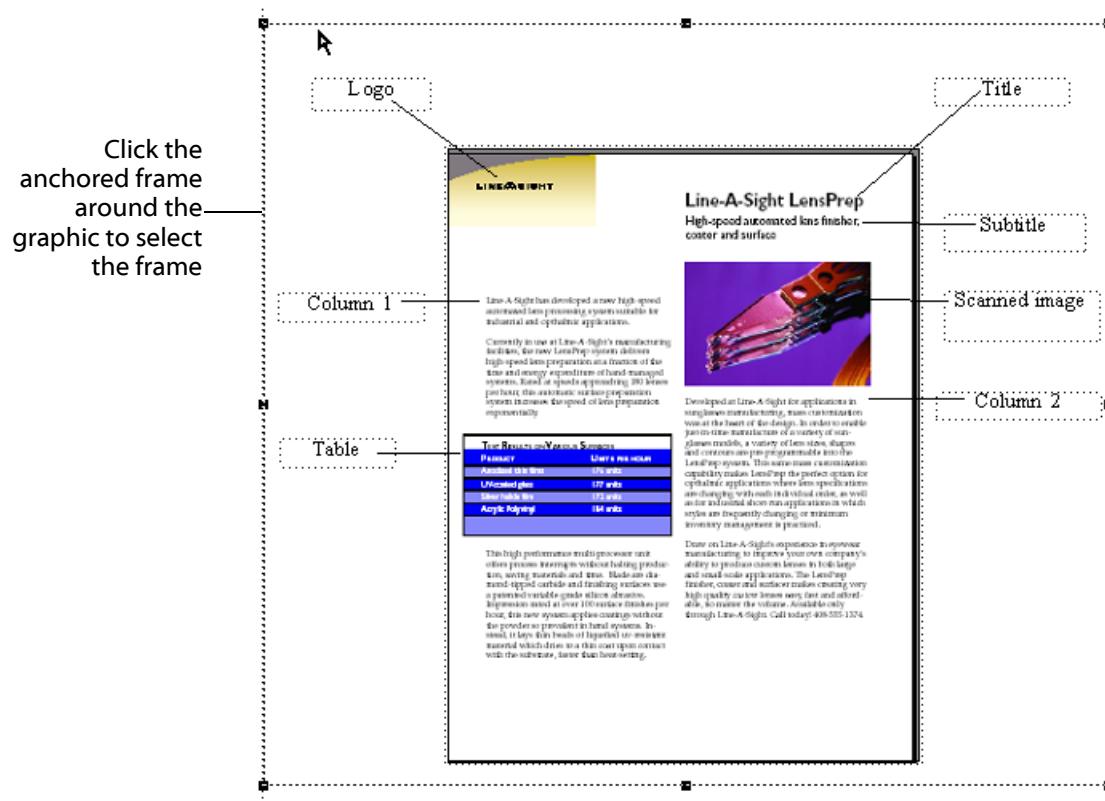
At the end of the exercise you should know:

- How to use FrameMaker to provide descriptive text for graphics that will be processed by devices like screen readers.
- Options for using graphic artifacts in, or eliminating them from, the logical flow of the document

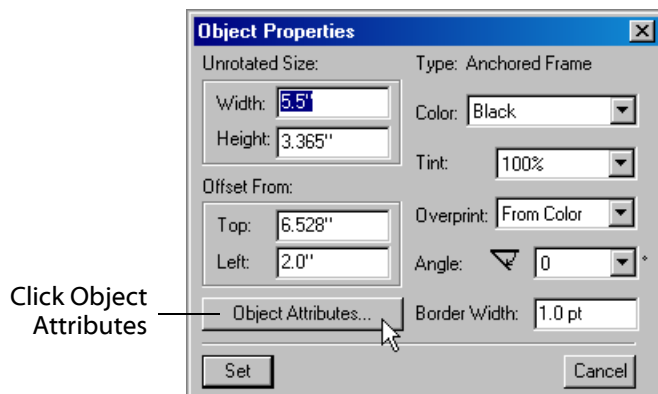
Using the Alternate Text Attribute

1. Open the sample document, **sample_fm_doc_1.fm**, in FrameMaker 7, and navigate to the graphic on page 5.
2. Click the anchored frame around the graphic to select it.

[Instructor Notes: Make sure they select the anchored frame surrounding the graphic, not the graphic itself.](#)

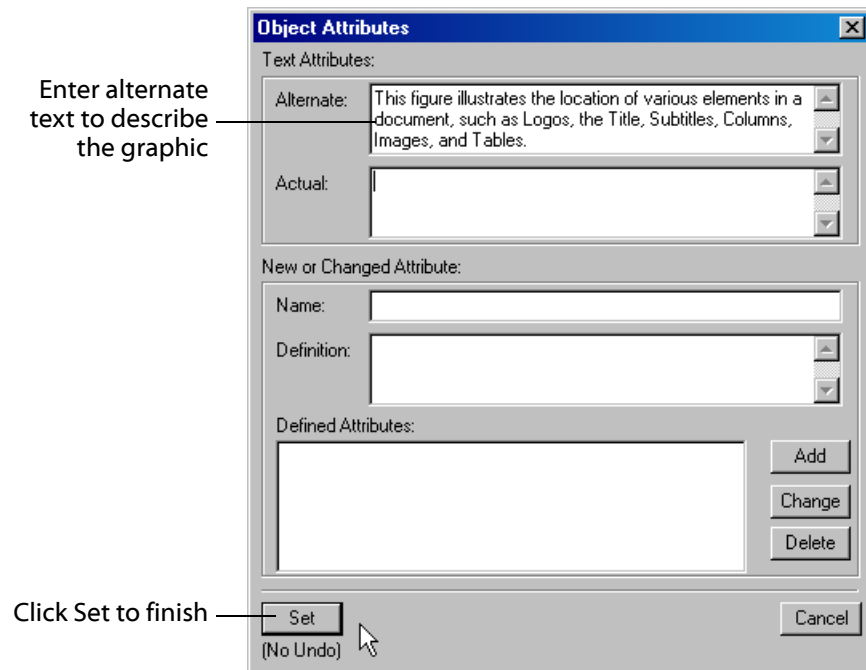


3. Select **Graphics > Object Properties** to bring up the Object Properties dialog box.
4. Click the **Object Attributes** button to open the Object Attributes dialog box.



5. Enter a description for the graphic on page 5 in the **Alternate** attribute field.

Use the description in the illustration or make up one of your own. Keep in mind that it should try to convey as much information as the graphic would convey to a sighted reader.



6. Click the **Set** button at the lower left when finished with the Alternate text.
7. Proceed the to the other graphics on pages 12 and 13.
8. Select the anchored frame around each, and use the **Graphics > Object Properties > Object Attributes** dialog to add descriptive text.

Write a description that conveys the meaning of each graphic, or use the descriptions in the [Making Graphics Accessible](#) section.

9. Save the file.

Adding Alternate Text for Context and Navigation

Suppose you decided it would be useful to make information about certain changes of context in a document—such as the beginning of a new topic section or an exercise, or the presence of an icon used for special notes and cautions—available to devices like screen readers. One way you could do that in the sample document would be to provide alternate text for the icons that mark the beginning of each section, and for icons that mark important passages in the document. In this section you will do this for the two icons on page 10 of the sample document:

1. Navigate to page 10 of the FrameMaker sample document.
2. Click to highlight the anchored frame around the Topic Heading graphic.



3. Select **Graphics > Object Properties > Object Attributes**.
4. In the **Alternative** attribute field, type something like: *Beginning a new topic*.
5. Click **Set**, then **Set** again.
6. Click to highlight the anchored frame around the Special Note graphic.



7. Select **Graphics > Object Properties > Object Attributes**.
8. In the **Alternative** attribute field, type something like: *This is a special note*.
9. Click **Set**, then **Set** again.

[Instructor Notes: Mention to students that they will test the results of these modifications in the next exercise.](#)

Handling Irrelevant Graphics

You saw in the last section how some visual decorations and markers can be used to provide alternate text for context and navigation purposes. But what about all the graphics in the document that are completely superfluous? And what about the 351 graphics with no alternate text that the Accessibility Checker found in the exercise [Accessibility Testing](#)?

Keep in mind that if you do not supply alternate text for a graphic, it is simply not processed by alternate devices like screen readers. Therefore, the simplest way to deal with irrelevant graphics is to do nothing with them: they will not be referenced in the logical flow of the document.

In regard to dealing with the many graphic elements flagged by the Accessibility Checker, here are some options:

- Turn off the Alternate Descriptions Provided option in the **Tools > Accessibility Checker > Accessibility Checker Options** dialog. The disadvantage is that graphics without alternate text will not be flagged at all, and you might miss a useful warning.
- The *recommended* way to deal with the Accessibility Checker's warnings about the several hundred graphic components without alternate text on pages 12 and 13 is to provide an Alternate Text value for the anchored frame surrounding the graphic elements. Refer to the section [Adding Alternate Text to Graphics](#) for the procedure. When you set a value for the Alternate Text attribute for a graphic frame, that value applies to everything in the subtree below it. Once you supply an Alternate Text value for the anchored frames around the graphics, the Accessibility Checker will cease its

complaints about the graphic elements contained in the frame. If you do not want to provide actual spoken alternative text for the graphic frame, you can supply a space character for the Alternative Text value.

Saving as PDF

Now that you have added alternative text to some of the graphics in the FrameMaker sample document, you need to recreate the corresponding tagged PDF file.

1. Save the sample document in FrameMaker.
2. Use the **File > Save As** command to open the Save Document window.
3. Select **PDF... (*.pdf)** from the **Save as type:** drop-down menu.

Make sure you have selected **PDF** as the **Save As type**. A message box appears asking, in effect, whether you want to save the file as **sample_fm_doc_1.pdf**.

4. Click **Yes**.

Another message box may appear asking if you want to overwrite the existing **sample_fm_doc_1.pdf** file. If so, click **Yes**.

5. Click on the **Tags** tab.
6. Make sure that the check box **Generate Tagged PDF** is selected.

You must do this to create an accessible PDF document.

Make sure that no paragraph formats are displayed in the Don't Include: window. If they are, click the **Get Defaults** button at the lower right to select all non-selected paragraph formats.

7. Click **Set**.



Exercise: Testing Alternate Text for Graphics

In order to do any of the exercises in this module, you must have Acrobat 5.x and FrameMaker 7.0 or higher installed on your machine, and you must have copied the sample files from **FMAccessibility.zip** to a folder on your machine.

The easiest and most direct ways to verify whether your alternate text is included in the logical read order are to:

- Use the Acrobat Tags Palette to verify that alternate text has been included in the tag structure.
- Save the document as accessible text and then search the text document for the alternate text to verify that it is there and in the right place.

Assuming that the alternative text is present in the PDF tag structure or in the accessible text file, we recommend that you also use a screen reader in a final test of whether your alternate text is spoken and in the correct sequence in the document.

Using the PDF Tags Palette

1. Open the **sample_fm_doc_1.pdf** file in Acrobat.
2. Click the **Tags** tab to open the Tags Palette in Acrobat.
3. Navigate to the graphic on page 5.
4. Use the Touch Up Object Tool to select the graphic.
5. In the Tags Palette, click on the Tags arrow to select the Tags menu, or right-click in the Tags Palette area to bring up the menu, and select **Find Element From Selection**.

This will cause the Tag corresponding to the graphic to be highlighted in the Tags Palette. Refer to the screen shot below.

[Instructor Notes: Make sure students understand step 5](#)

PDF tag for the Heading Section 508 and Electronic Documents

PDF tag for the graphic on page 5

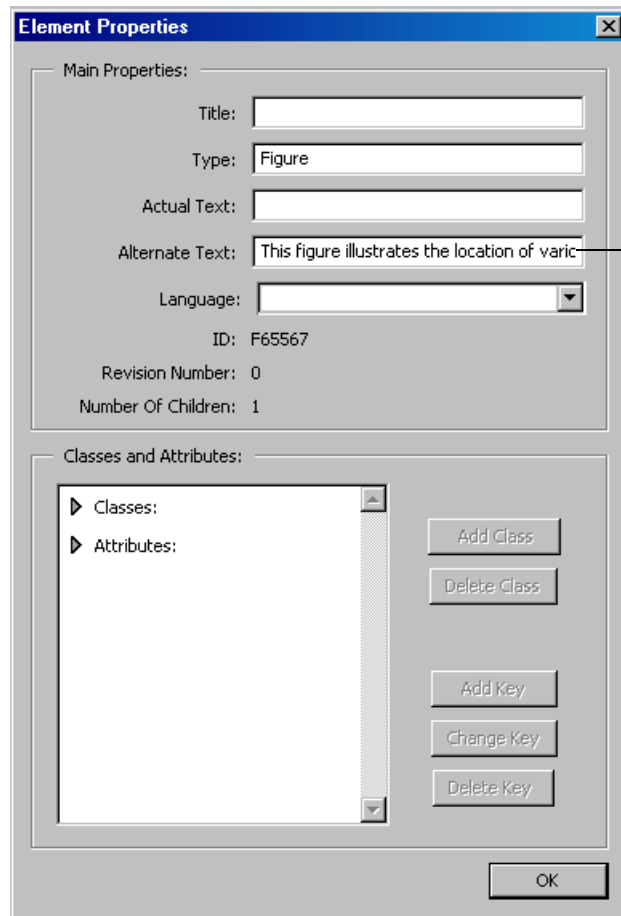
Make sure Associated Content Highlighting is off

Select Element Properties to see attributes for the <Figure> tag

6. Right-click the **<Figure>** tag for the graphic on page 5 and select **Turn Off Associated Content**.

If you don't do this, you will not be able to access the rest of the menu items.

7. Right-click the **<Figure>** tag for the graphic on page 5 and select **Element Properties**.
8. Verify that the **Actual** or **Alternate** text you added in FrameMaker are displayed in their respective fields.



The Alternate attribute declared for the figure's anchored frame in FrameMaker should appear here in the Alternate Text field of the PDF <Figure> Tag's Element Properties.

Saving as Accessible Text

1. In Acrobat, select **File > Save As**.
2. Choose **Save as type: Text (Accessible)** and name the file **sample_fm_doc_1.txt**.
3. Overwrite the existing file if necessary.
4. Click **Save**.
5. Open **sample_fm_doc_1.txt** with a text editor and use either **Edit > Find** or **ctrl-F** to search for instances of the alternate text you added to the FrameMaker anchored frames containing the graphics.

Pay attention to whether the alternate text appears in the correct logical sequence relative to the surrounding text.

Instructor's Note: the FrameMaker Distiller places a line feed character at the beginning of the Alternate attribute value string, which sometimes displays as a Ÿ in the text, or as a square in the Alternate Text field in the Element Properties dialog. This is a recognized bug that should be corrected by the time these materials are in your hands, but, in any case, it should not affect the way screen

[readers process the text.](#)

Using a Screen Reader

1. Start the installed screen reader, or have the instructor do so.
2. Use the Bookmarks Palette to navigate to a page containing a graphic with alternate text.
3. Start screen reader processing and note whether the alternate text for your graphics is read, and in the correct logical sequence.



Working with Hypertext Links and Cross References

If you follow correct formatting procedures when you create hypertext URL links and cross references in FrameMaker, the Acrobat Distiller does an excellent job of converting the link object references and the text of the link to the PDF document. The hypertext or cross reference link and its text are represented in the PDF tag structure in the correct sequence. A screen reader will usually announce the presence of a link when it is encountered, and the user can execute a command to jump to the reference target in the same document or a different document. Hypertext links to URLs function in much the same way. If need be, refer the FrameMaker 7.0 Help documentation for guidelines on how to create cross references and hypertext links to URLs.

There is a remaining accessibility issue in the sample document, one that the Accessibility Checker flagged, which is that two hypertext links exist not in the main text flow in the Body Pages, but in the Master Pages. These links are not included in the PDF tag structure and will not be accessible without further work. Although the decision of whether or not to include in the logical flow what are, in effect, boilerplate cross references is a design decision to some extent, assume in this case that you are legally required to make statements about privacy and terms of use policy accessible to visually impaired readers. The next exercise will cover one way of doing this using a hidden hypertext link to a URL. The procedure would be similar for creating a hidden cross reference link to another area of the document, or to an external document.



Exercise: Using Cross References

In order to do any of the exercises in this module, you must have Acrobat 5.x and FrameMaker 7.0 or higher installed on your machine, and you must have copied the sample files from **FMAccessibility.zip** to a folder on your machine.

In this exercise, you will make the two hypertext links at the bottom of the first page of **sample_fm_doc_1.fm**, [Online Privacy Policy](#) and [Terms of Use](#), accessible to screen readers.

If you save the **sample_fm_doc_1.fm** file as a PDF file, and click either of the two hypertext references at the bottom of the first page ([Online Privacy Policy](#) or [Terms of Use](#)), the corresponding URL is displayed in your browser. However, the hypertext links do not exist in the PDF tag structure and therefore are not available to visually impaired users.

The reason for this is that these two references were defined in the Master Pages of the sample FrameMaker document. When a FrameMaker document is converted to PDF, elements from the Master Pages are treated as artifacts. One effect of this is that certain elements, such as page numbers and running heads which are not particularly relevant to screen readers, are eliminated from the text flow. Unfortunately, any relevant content elements that are placed in Master Pages, which is sometimes a very useful technique, are not included in the logical read order. You will fix this problem for the sample document by placing the hypertext links in the main text flow without changing the visual character of the existing document.

1. Open **sample_fm_doc_1.fm**.
2. Select **View > Master Pages**.

You cannot access links that have been declared in Master Pages from the Body Page view.

At the bottom of the last Master Page (the master page labelled "First") are the hypertext links for **Terms of Use** and **Online Privacy Policy**.

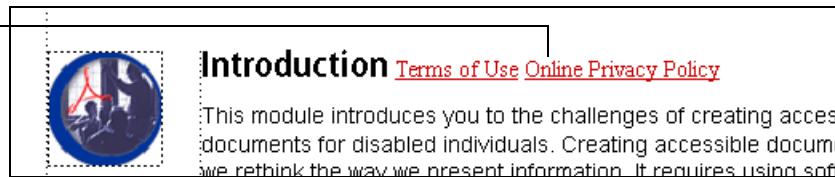
3. Highlight the [Terms of Use](#) link and select **Copy**.
4. Select **View > Body Pages**.
5. You want to place the copied link somewhere within the main text flow in a position that is functionally equivalent to where it now appears as a footer defined on a Master Page. This is a judgement call to some extent:
 - The hypertext links appear on page one, indicating the authors intended the information to be available immediately to readers.
 - You want to add the links to an existing paragraph, rather than creating a new paragraph for it, in order to produce minimal visual disruption of the document. Later, you will hide the text from view.

A relatively safe place to put them would seem to be at the end of the Introduction heading.

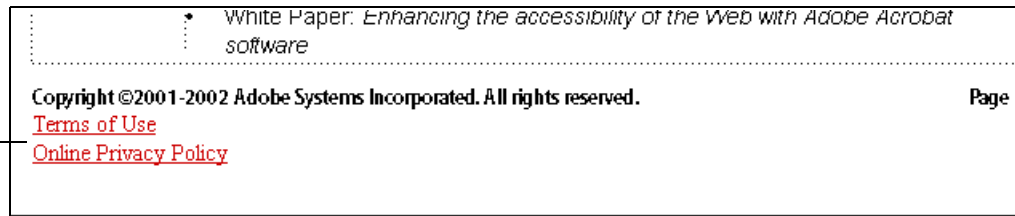
6. Put the cursor on the same line and just after the Introduction heading.
7. Type one space and then paste the **Terms of Use** link there.
8. Repeat steps 2. through 9 for the **Online Privacy Policy** hypertext link in the Master Pages.

You should now see copies of the two hypertext links on the same line as the Introduction heading.

Copies of the
hypertext links in
the main text
flow in the Body
Pages



Hypertext links in
a footer in the
Master Pages



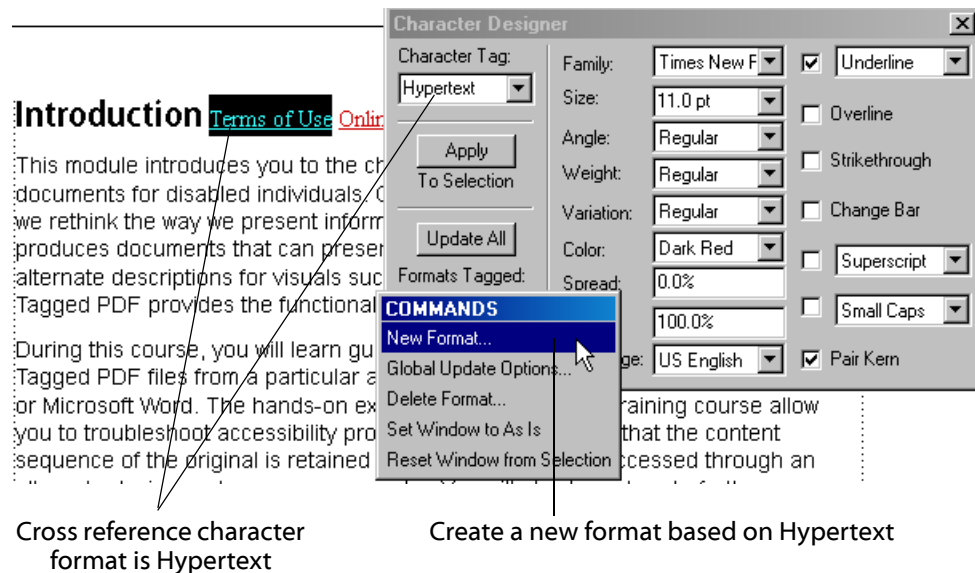
Hiding Text Added for Accessibility

Although you have added accessible text for the links, you want to be able to provide accessibility to them without disturbing the document visually. Since only one page, page one, uses the master page with the hypertext links, the correct design decision probably would have been to put them in the Body Pages to begin with. Given the present situation, however, you can hide the copied links so that sighted readers will not be aware of them and visually impaired readers will be. One way to do this is to create a non-visible character format and apply it to the copied hypertext.

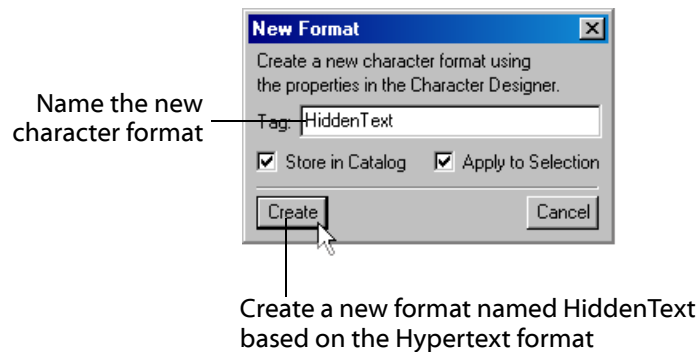
1. Highlight the copied **Terms of Use** link.
2. Select **Format > Character > Designer** to bring up the Character Designer dialog.

Notice that the character format that the cross reference uses is Hypertext.

3. Click on the **Commands** drop-down menu and select **New Format**.



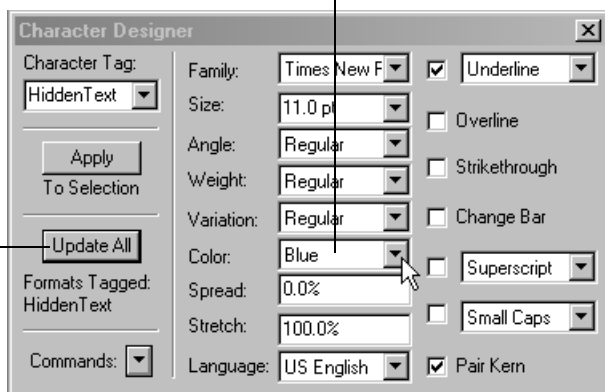
4. Enter the name **HiddenText** in the Character Tag field.



5. Click the **Create** button.
6. In the **Character Designer** dialog, change the value in the **Color** drop-down list to something distinct, like **Blue**.

Select a unique color to use for HiddenText during development. Prior to delivery, you will change the color to White.

Click Update All to apply the color change to all HiddenText in the document.



7. Click the **Update All** button.

- If the **Terms of Use** link is still highlighted, it changes color when you click the **Update All** button.
- If it is not still highlighted, click on it now, and then click on the HiddenText tag in the Character Catalog to apply that character format to the link. Do the same for the **Online Privacy Policy** link.

8. You will use a visible color for HiddenText during development of the document. Prior to delivery of the document, open the Character Designer for the HiddenText character tag and change the color of its font to White. Clicking **Update All** will then hide all HiddenText passages in the document.

9. Select **Format > Character > Designer**.

10. Select **HiddenText** from the **Character Tag** drop-down list.

11. Select **White** from the **Color** drop down list.

12. Click **Update All**.

The hypertext links you copied to the main text flow should now be invisible in the document.

13. **Save** the FrameMaker file.



There are certain drawbacks to this technique. If the user changes the text color and background color, which is an option recommended for users with low visual acuity, the text will become visible. At this time, there is no way to select a truly invisible character color. Note, also, that sighted users who are tabbing through the elements of the page will tab to the invisible elements, which may be confusing.

Instructor Notes: On occasion, it has been noticed that the text color changes are not correctly exported to the PDF file until the FrameMaker file is saved and

[reopened. The text appears hidden in FrameMaker but is visible in Acrobat. If you run into this problem, try that solution.](#)

14. Save the document as Tagged PDF.
15. Overwrite any existing PDF file.
16. In the PDF Setup dialog, **Tags** tab, make sure **Tagged PDF** is selected and that all paragraph formats have been selected in the **Include Paragraphs** window. (See [Saving FrameMaker 7 Documents as Tagged PDF.](#))
17. Open the PDF file.

You will observe the following:

- The links for **Terms of Use** and **Online Privacy Policy** that originated from the footer in the Master Page of the original FrameMaker document are active, and clicking them brings up the associated document.
- The links you copied into the main text flow of the FrameMaker file are also active, if you can find them after the Introduction text, but they are not visible. The Acrobat cursor, however, will change to a pointing finger icon when it is over those areas.

18. Save the PDF file as accessible text.



Text (Accessible) is only available with Acrobat 5.0.5 or greater running under Windows. You can save the PDF file as Text (Plain), but Alternate Text for graphics may or may not be displayed. Actual Text is displayed in both Text (Accessible) and Text (Plain) files.

19. Open the text file with a text editor and search for the terms **Terms of Use** or **Online Privacy Policy**.

You will find them at the end of the text for the Introduction heading, where you put them. This means that devices like screen readers will be able to access the hypertext in the links.

Further Testing

You can verify that the links and their object references have been correctly transferred to the PDF tags structure by opening the Tags Palette and searching for the **<Link>** tags associated with the cross references you copied to the Body Pages. You will find them nested under the first **<_Heading1>** tag.

You can also verify that the links have been correctly transferred to the PDF document and that they are active by starting a screen reader and reading just past the Introduction heading on page one. The screen reader should identify each link as such and read the text. You can then use the screen reader commands to access the URL from your browser.



Supplemental Exercise: Using Hidden Cross References for Navigation

[Instructor Notes: Use this exercise if you need to fill time or for the speedy ones in the class.](#)



You may wish to try this exercise if time permits.

Cross references can sometimes be used in the service of easing the difficulty of navigating through a document with a screen reader. They can be used to approximate some of the visual cues about the structure of the document and the meaning of certain parts of the document that fully sighted readers use on an almost automatic basis. For example, copyright notices, newspaper mastheads, disclaimers, license agreements, and other such boilerplate information, tend to be immediately recognized by sighted readers for what they are, and then in general ignored immediately unless the reader has a specific reason for closely attending to the information. Sighted people, in other words, can use cues available in the visual layout of the document to selectively ignore uninteresting sections of it. Screen readers, on the other hand, will read everything in the logical flow with no discrimination. The screen reader user can sequentially page through lines of text until a section of no interest is finished, although it may not be clear where the end of the section lies.

One way cross references can be used is to provide functionality that is somewhat equivalent to being able to visually recognize and skip over a section of a document that is not of interest. As you saw in the previous exercise, if the text in such a cross reference, for example, *Select this link to skip over copyright information*, is hidden, it will still exist in the PDF tag hierarchy and will be accessed by devices such as screen readers. The target of such a cross reference should be the first paragraph beyond the section to be skipped.

1. There are no blatantly obvious candidates for skipping in the sample document. As an exercise left solely to the reader however, create a hidden cross reference at the beginning of the [References](#) section which will allow readers to skip the reference information.
2. The target source for the cross link should be the **Heading2** paragraph *File to Download for Exercises in this Module*, which is the next section in the document after the *References* Heading.
3. Test the cross reference link, then apply the **HiddenText** character format to it.
4. Save the document as Tagged PDF and test the functionality of the links.



Summary

You have been introduced to the following topics in this module:

FrameMaker 7 and Tagged PDF

Tagged PDF allows information about the structure of a document, as well as its contents, to be used by accessibility devices. When a FrameMaker 7 file is Exported to PDF, a Tagged PDF hierarchy is created that represents the various parts of the document. The degree to which the original FrameMaker document has been formally structured and formatted will affect the quality and structure of the generated PDF tag hierarchy. The PDF tag hierarchy determines the order in which the elements of the PDF document will be processed by devices like screen readers. Structuring the original document usually begins with an analysis and specification of how its logical read order should flow.

The FrameMaker Acrobat Distiller process does a very accurate and thorough job of converting FrameMaker documents to Tagged PDF. Alternate text for graphics must be supplied by the FrameMaker developer, and hidden cross references can be supplied for navigational purposes. FrameMaker 7 allows the developer to minimize if not eliminate the amount of post-PDF processing required to accurately produce an accessible document. Post-processing changes to the PDF tag structure cannot be saved and must be manually reapplied each time the original document is reconverted, for example, after an update. The advantages of avoiding post-PDF conversion changes to the Tags structure should be apparent.

Controlling Logical Read Order

As easy as it is to create Tagged PDF from FrameMaker, you will still encounter problem areas, increasing with the complexity of the document. You will need to make decisions about the relevance of certain visual embellishments, and consider ways to provide meaningful alternate descriptive text for graphics. Although the field is still relatively new, useful resources regarding document accessibility design and implementation are available, and we recommend you begin with the [References](#) on the first page of this document.

To a great extent, the workflow is a develop-and-test iteration until the test criteria are met. At minimum, the test criteria should always specify that the document flows logically and includes text alternatives for relevant graphics. Testing should be done using multiple testing methods rather than relying on one; ideally part of the testing should be done using a screen reader.

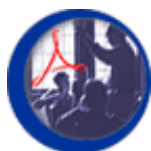
Testing for Accessibility

The PDF tag hierarchy is displayed in the Acrobat Tags Palette. Each tag in the hierarchy contains a child tag until an end node is reached, which represents a piece of the actual content of the document. Various methods are available for testing the logical flow inherent in the PDF tag structure and for analyzing the tag structure itself:

- Associated Content Highlighting can be used to see if logically connected areas of the document correspond to the PDF tag order.

- The PDF file can be Saved As Text (Accessible). The order in which the contents of the document are displayed in the text file represents the document's logical read order.
- A screen reader can be used to check logical flow.
- The Accessibility Checker can be used to find non-tagged document content and to highlight potential accessibility problems in the document.

Module 3: Using the Tags Palette



Introduction

This module introduces various kinds of accessibility problems that you may face in your Tagged PDF. While we highly recommend that you fix as many problems as possible in the authoring software, there may be cases where this is not possible. We encourage you to keep the number of repairs that you need to make using the Acrobat Tags palette to a minimum to avoid having to redo them with each new revision of a document.

Instructor Notes: Modifications made directly to the PDF Tag structure in Acrobat cannot be saved. Each time the PDF document is updated and re-generated from the original authoring tool, for example, InDesign, FrameMaker, or MS Word, manual changes made directly to the PDF document must be reapplied. Acrobat 5 provides a great deal of flexibility in regard to modifying the PDF Tag structure, but make sure students understand the advantages of avoiding post-PDF conversion changes to the Tags structure when possible.

This module is organized as a series of exercises to allow you to get the hands-on practice that you need with the Tags palette.

Learning Objectives

At the end of this module, you will be able to:

- Understand the difference between logical read order versus page content order
- Rearrange PDF tag elements to fix logical read order
- Modify the reflow of a document to fix page content order
- Create new tag elements to add explanations or fix problems with links or text in existing tags
- Use Tag palette commands to find untagged content and dynamically create tag elements

References

- *How to Create Accessible Documents*
- *Advanced Techniques for Creating Accessible Adobe PDF Files*
 - <http://www.adobe.com/products/acrobat/pdfs/CreateAccessibleAdvanced.pdf>
- *Adobe® Acrobat® Help (online document included with Acrobat 5)*

Contents

Topics	Exercises
Logical Read Order and Page Content Order	Fixing Logical Read Order Problems
	Fixing Text Reflow Problems

Topics	Exercises
Fixing Content Problems	Creating New Tag Elements
Converting Missing Content	Converting Missing Content to Tags

File to Download for Exercises in this Module: TagsPalette.zip

TagsPalette.zip contains the following files:

- sample_text_reflow.pdf
- UserGuideStart.pdf
- UserGuideSolution.pdf
- MktPlan.pdf
- MktPlanSolution.pdf



Logical Read Order and Page Content Order

Tagged PDF is created when a source file is converted, exported, or Saved As PDF. The specific method for generating the PDF file depends on the authoring tool used for the source document.

You will work with Tagged PDF documents in the exercises that follow. First, however, you should be aware of two related but distinct characteristics of accessible documents, *logical flow*, and *text reflow*:

- We use the terms *logical flow* and *logical structure*, and occasionally *logical read order*, interchangeably.

The physical manifestation of *logical flow* is the order in which a screen reader speaks the content elements of the document. The order of content elements in a document's logical flow does not necessarily have to be the same as the visual flow or placement of elements in the document.

- The hierarchical order of the Tagged PDF structure determines the logical flow of the document. Consequently, we use the term *Tagged PDF structure* to mean almost the same thing as logical flow or logical structure.
- We use the term *text reflow* or *reflow* to refer to the order in which content elements of the document are wrapped when you enable Reflow in Acrobat. Once Reflow is enabled, Acrobat wraps and fits the content of the document to the document window when it is reformatted to different proportions, or when the display is magnified.

Reflow allows a PDF document to fit the reading area whether that be the user's desktop or handheld PDA. The Reflow view is active only when the PDF document is Tagged PDF. Tables and Figures do not reflow, but rather scale to fit the reading area. The order of elements in a reflowed document does not necessarily have to be the same as the visual flow of elements in the original document

- Text reflow is conceptually and physically separate from logical flow and tagged structure. The two do not affect each other; you must handle each separately.
- Text reflow order is important in regard to providing accessibility to people who are visually impaired but not functionally blind. These users can use the text magnification and reflow capabilities that environments like Acrobat 5.0 provide. Reflow capabilities are important for accessibility because text magnification by itself obscures visual context and forces users to scroll widely about the document in order to follow its logical flow. With reflow capabilities, magnified text of any size is reflowed properly into the available space.



The Acrobat Reflow feature works reliably for tagged Adobe PDF documents that contain Roman language text. It does not work reliably for tagged Adobe PDF documents that contain Asian language text.



Reflowing a document is for on-screen viewing only. You cannot save or print the reflowed state of a document.

The table below compares and contrasts logical read order and page content order. Note that if you want to insure that your PDF document is accessible to the blind via screen readers, you must verify the document's logical read order. If you want readers who are visually-impaired, but not blind, to be able to view a PDF document using a screen magnifier, you must verify the page content order. In general, you should do both.

	Logical Read Order	Page Content Order
Used by screen readers	✓	—
Used for reflow	—	✓
Used for repurposing (saving to other formats)	✓	—
How to view	Tags palette	Touchup order tool or select reflow tool
How to edit	Reorder tag elements in the Tags palette	Select desired text with the touchup order tool
How to verify	Select File > Save As , choose Text (Accessible)	Select View > Reflow

Fixing Logical Read Order

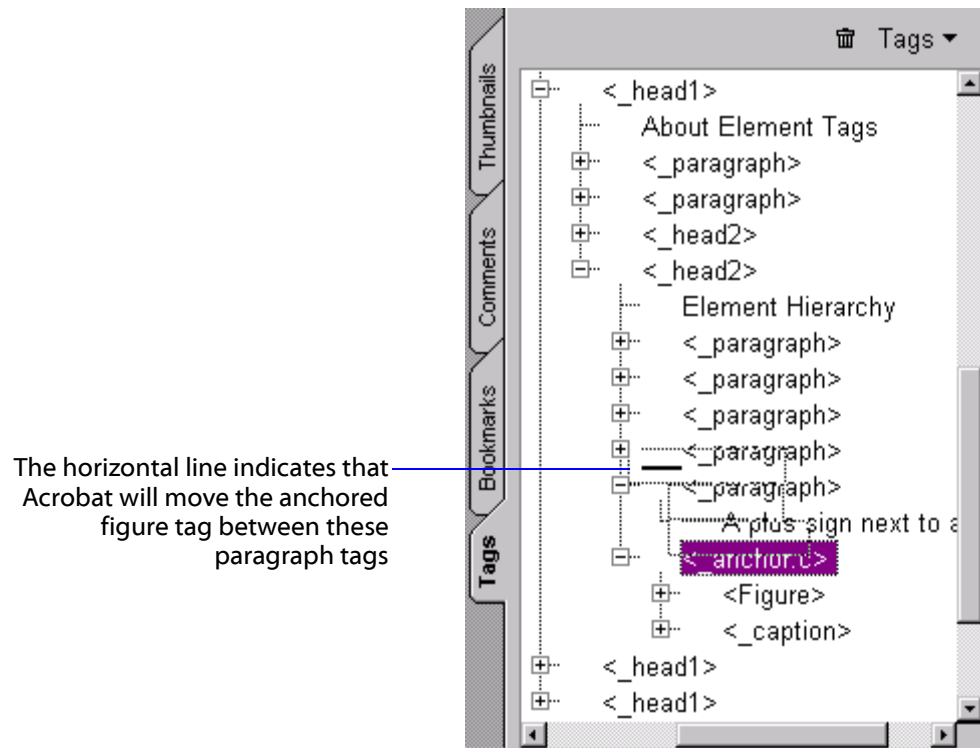
Instructor Notes: The following paragraphs and screen shots are an overview and introduction to using the PDF Tags Palette to manipulate the PDF Tag structure. Students will have a chance to do this in the next exercise. If you have access to a screen projector or display that can be seen by the entire group, you may want to demonstrate the basic principles to the class as they are introduced below.

In the first exercise in this module, you will use the PDF Tags Palette to correct some problems with the logical read order of a document. The following is an overview of the basic procedures:

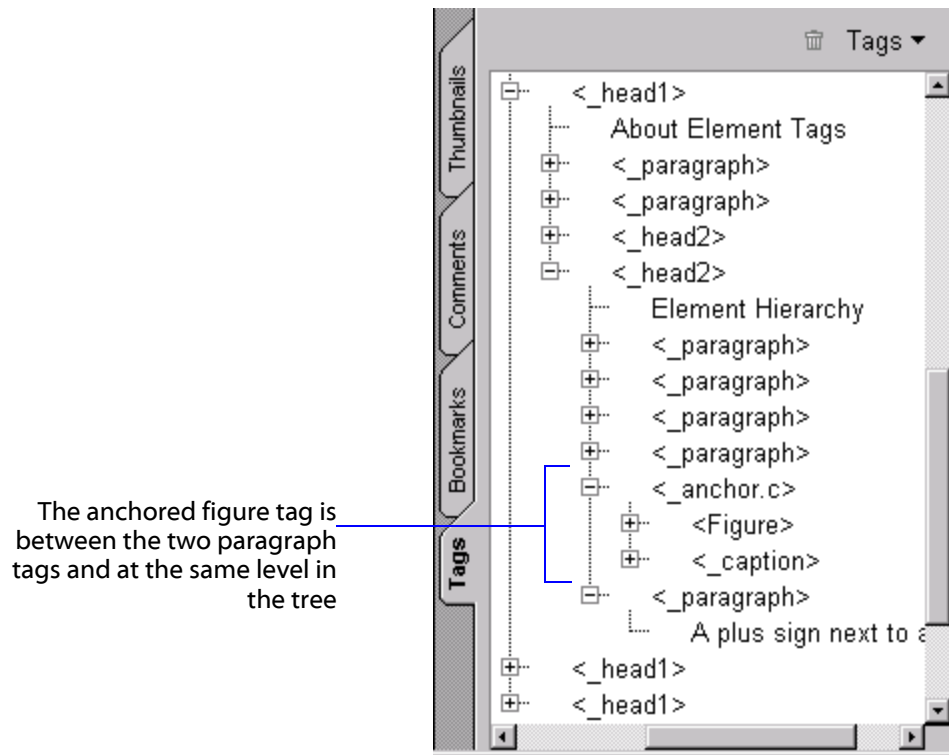
When you open a PDF document in Acrobat 5, a *Tags* tab is displayed in the Palette Window at the left. If the loaded document is a Structured PDF file, clicking on the Tags tab opens the Tags Palette and displays the document's PDF Tag structure. At first, you may see only one element, the *Tags Root*, which can be expanded to reveal the entire PDF Tag hierarchy. You can also reorder the Tag elements in the Tags Palette, which in turn affects the logical flow of the document without affecting its physical appearance.

The interface works in much the same way as moving files in the Windows Explorer. Using the left mouse button, you can select a tag element with the

mouse. You can move selected tag to different locations within the structure tree by keeping the left mouse button depressed. Once you reach the desired location, release the mouse button. You can also select more than one tag, provided that they are sibling tags in the structure tree. By using the Shift key and the left mouse button, you can select contiguous tags. You can use the Ctrl key and the left mouse button to select non-contiguous tags. When you move a tag or group of tags, the insertion mark is shown as a horizontal line.

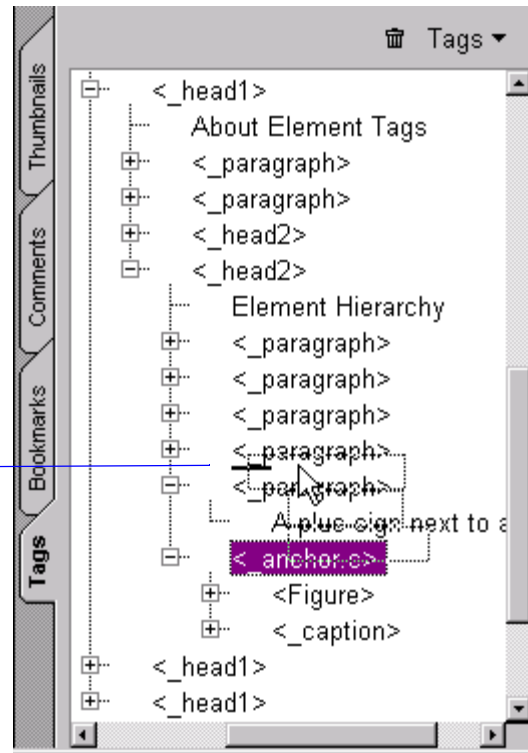


When you release the left mouse button, in the diagram above, Acrobat inserts the **<anchor.c>** tag and its child tag elements between the two paragraph tags. The resulting structure tree is shown below.



You need to pay attention to the horizontal position of the insertion mark. It indicates the level within the structure tree where Acrobat will insert the tags. In the diagram below, the insertion mark is even with the paragraph tag above it. This indicates that Acrobat will insert the selected tag as a child element of the paragraph tag instead of a sibling element. The difference is subtle but critical. Compare the two insertion examples and the results. Then try some experiments of your own.

The horizontal position of the insertion mark indicates that Acrobat will insert it as a child under the paragraph tag

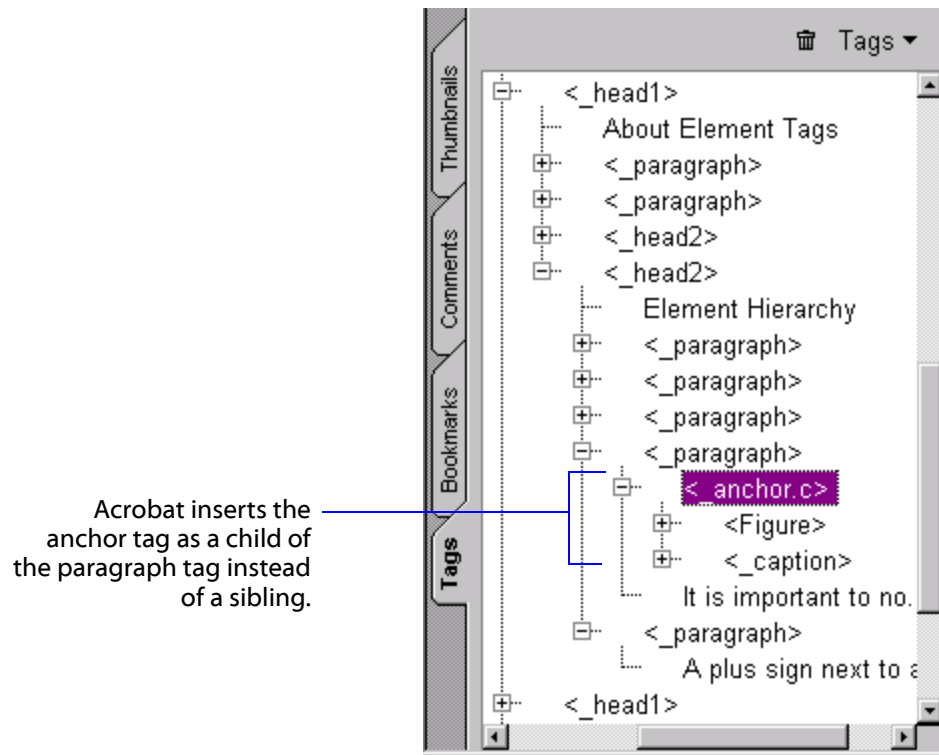


The diagram below shows the result. Acrobat has inserted the **<anchor.c>** tag as a child of the **<paragraph>** tag.

Instructor Notes: When Turn On Associated Content Highlighting is active, no other commands from the Tags menu can be selected.




If you see a circle with a slash through it instead of the insertion mark, Acrobat cannot move the selected tag elements to that location. If you see this symbol even in areas of the tree where the tag can legitimately belong, you are in content association mode. To check if you are in content association mode, right-click the mouse in the Tags palette to bring up the pop-up menu. If you only see one menu item, **Turn Off Associated Content Highlighting**, click it to exit content association mode.



Finding the Associated Element for Document Content

[Instructor Note: Demonstrate the following procedure to the class if possible.](#)

In the second module in this course, you learned how to step through the structure tree and view the associated document content for each element. There may be times when you are looking at the document and want to find the associated tag element for a particular paragraph, diagram, and so on. Rather than laboriously stepping through the structure tree, you can reverse the process by first

highlighting the desired page content, with the Text Select tool . Then, open the Tags drop down menu at the top of the Palette Window, and select **Find Element From Selection**. Acrobat expands the structure tree as appropriate and then highlights the associated element.



Exercise: Fixing Logical Read Order Problems

In order to do this exercise, you must have Acrobat 5.05 installed on your machine. You must also have downloaded and unzipped the file **TagsPalette.zip** from the Accessibility training Web site.

In this exercise you will:

- Use the Tags palette to find a paragraph's associated element in the structure tree
- Use content association to examine the logical read order
- Rearrange tag elements in the structure tree to fix the logical read order

At the end of the exercise you will be able to:

- Find the associated tag element for any document content
- Modify reading order by moving tag elements
- Understand that the insertion mark indicates the level of the structure tree
- Verify the revised reading order

You will work with and modify the sample file **UserGuideStart.pdf**. You can find an example of the correctly ordered document in **UserGuideSolution.pdf**.

1. Make a copy of **UserGuideStart.pdf** and open it in Acrobat.

This document, which is a short user guide, has problems with logical ordering. You will fix the logical read order by rearranging the Tag elements in the PDF Tags Palette. Since the goal is to understand how the PDF Tag Palette works, ignore for a moment that it would be preferable to fix the ordering problem by editing the original source document.

2. **Open** the Tags palette.
3. Go to page 4 of the document. Page 4 is the first page of Chapter 2.

You will see a several text boxes that were used to create a flowchart. A quick way to jump into the structure tree is to click on the Text Select tool and use the mouse to select the text in the document pane. Use the procedure below.

- a. Use the **Text Select** tool to select a portion of the text in the first text box. For example, highlight the text "Get the current list of students...".
- b. Click on the drop down **Tags** menu, located at the upper left of the Palette Window. Select **Find Element From Selection**. This opens up the structure tree and positions the cursor on the right element.
- c. Use the **Text Select** tool to select a portion of the text in the last text box. For example, highlight the text "Back up last year's data...".
- d. **Click** on the drop down **Tags** menu, located at the upper left of the Palette Window. Select **Find Element From Selection**. Notice that the last text box is located before the first text box in the PDF Tag hierarchy. This

means that the text in the last text box will be read by a screen reader before the text in the first text box, and as such will definitely not convey the intended meaning of the chart.

4. **Click** on the **Tags** drop down menu, or left-click on one of the elements in the Tags Palette, and select **Turn On Associated Content Highlighting**. Click on each of the **<TextBox>** elements in the Tags Palette that are between the last and the first text boxes you selected in Step 3, above. Notice that the text boxes are placed in the PDF Tag hierarchy in reverse order of what should be their logical order.
5. Use the following procedures to rearrange the text box elements in the structure tree so they are in the correct logical order:

Instructor Notes: Remind students that they will not be able to move Tags Palette elements while Associated Content Highlighting is On.

First, note the positions in the Tags hierarchy of the first and last of the five **<TextBox>** elements in the chart that you are going to rearrange.

Open the **Tags** drop down menu and select **Turn Off Associated Content Highlighting**.

If you have expanded the tags for the first and last text boxes, you may find it easier to collapse them again before moving their positions. Click the parent tag's minus sign to do so.

Click on the last **<TextBox>** Tag in the group of five to select it.

Instructor Notes: There is an extraneous, non-printing <TextBox> element toward the top of the chart's border. If anyone asks, it is a tag for an empty text box whose origins are unknown. It will not be read by a screen reader and can be ignored for our purposes here. However, students may confuse it in the Tags Palette with the first of the five actual text box tags, and you may need to point this out to them.

Holding down the **left mouse button**, move the last **<TextBox>** Tag upwards in the hierarchy to a place just above the first **<TextBox>** Tag in the group of five. Move the cursor to the left as far as you can without causing the cursor icon to change from a horizontal bar to a circle with a line through it. **Release** the left mouse button to place the Tag in the new position.

Note that if you do not place the Tag as far to the left as you can, it will end up as a child element of the Tag above, which is not what you want. If that happens, grab it again and move it to the right place. The tricky thing about moving tag elements is to make sure they are inserted in the right level in the structure tree. It is very easy to wind up with elements as child elements instead of sibling elements. Pay attention to the horizontal position of the insertion mark. You may have to experiment a few times to get it right.

6. If the tags structure gets hopelessly messed up, close the document, without saving it, reopen it and start again. Also make sure the text box elements are in the same order as in the document pane.
7. **Save** your changes.

8. Test your changes by saving the file as **Text (Accessible)**.
9. Open the text file with an editor and search for a portion of text from one of the text boxes. Check to see that the sequence of text from all of the text boxes flows logically as it should.



Exercise: Fixing Text Reflow Problems

In order to do this exercise, you must have Acrobat 5.05 installed on your machine. You must also have downloaded and unzipped the file **TagsPalette.zip** from the Accessibility training Web site.

In this exercise, you will:

- Reflow a sample PDF document and note various problems with reflow order.
- Use the touchup order tool to fix reflow problems.
- Test the results.

At the end of the exercise you will understand how to control the reflow order of a document, and will understand how text reflow differs from logical order.

Remember, you need to find and fix reflow problems if you want your PDF document to be comprehensible to people who need to use screen magnifiers.


Instructor Notes: Be sure to point out to students who are using InDesign as their authoring tool that text reflow in Acrobat can be specified and controlled in the original InDesign document. This is preferable to fixing the text reflow as a post processing step in Acrobat for the same reason that it is preferable to fix logical flow in the original authoring tool, namely that changes made to text reflow order in Acrobat will be overwritten if the PDF document is re-generated from the original source. The last exercise in the InDesign module covers controlling text reflow in InDesign. It is a supplemental module, so not all InDesign students may have had time to go through it.

Examining Reflow Problems

1. Open the file **sample_text_reflow.pdf** using Acrobat 5.

This file was created by exporting an InDesign document to PDF format. Note the normal sequence in which the elements of the document are read:

- title
- left-margin hanging text and paragraph heading, or vice versa
- column one of paragraph one
- column two of paragraph one, and so on.
- Note that the main graphic is located toward the bottom of the page.

2. Find the reflow tool  in the toolbar and click it.

The contents of the page reflow sequentially to fit the window. The document contents are reformatted to fit the window after the user changes the window size or font magnification.

Notice a number of discrepancies in the way the text is sequenced when reflowed as compared to the original, as shown in the following figure. To name a few:

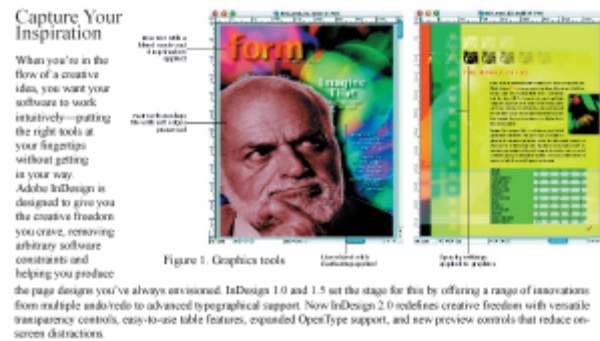
- The heading for the first paragraph is the lead item

- The multi-column paragraph on page one is split by the graphic from the bottom area of the page, the heading from the second section, and the hanging text in the left margin.
- The Title is at the bottom of the page.

Original layout
sequence



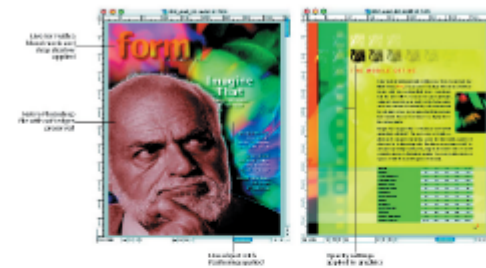
Next screen shot is
of the document
after text reflow



New Feature Highlights

Adobe InDesign 2.0 software sets new standards for professional layout and design, delivering the creative tools you've always wanted. How do we know? Because you—and designers and print professionals like you—use it to define what a page layout application should really be. With InDesign 2.0, you can:

- **Capture your inspiration.** InDesign 2.0 introduces groundbreaking creative features that transform how you approach page design. For the first time, you can create editable drop shadows, feathering, and transparency directly in a desktop page layout application. You can even place Adobe Photoshop® files with transparency backgrounds and soft edges. In addition, InDesign 2.0 takes the pain out of creating tables, so you design. Expanded OpenType® font support and new preview controls complete the new creative features in InDesign 2.0.
- **Be the master of deadlines.** InDesign is packed with intuitive enhancements that streamline elaborate design tasks, saving you time and freeing you to be more



When you're in the flow of a creative idea, you want your software to work intuitively—putting the right tools at your fingertips without getting in your way. Adobe InDesign is designed to give you the creative freedom removing arbitrary software constraints and helping you produce.

Setting new standards for professional layout and design

Capture Your Inspiration

creative. Now, with InDesign 2.0, you can prepare long documents as easily as you would short documents, including generating tables of contents and detailed indexes. The enhanced paragraph composer (formerly a composer) automatically sets optimal line breaks for each paragraph. An efficient new printing interface delivers precise, reliable results. And performance enhancements ensure that you can work more productively with

- **Work more efficiently through tight Adobe integration.** InDesign 2.0 integrates better with other leading graphics applications—Adobe Photoshop, Illustrator®, and Acrobat®—than any other page layout application. It shares commands, tools, palettes, and keyboard shortcuts with these applications, as well as core Adobe technologies such as the new Adobe Color Engine included in InDesign 2.0, Photoshop 6.0, Illustrator 9.0/10, and Acrobat InDesign 2.0 exports Acrobat 5.0 PDF files. The Japanese and Roman versions of InDesign 2.0 can exchange version 2.0 files because of their common file format. InDesign 2.0 also offers native support of the Mac OS operating system.



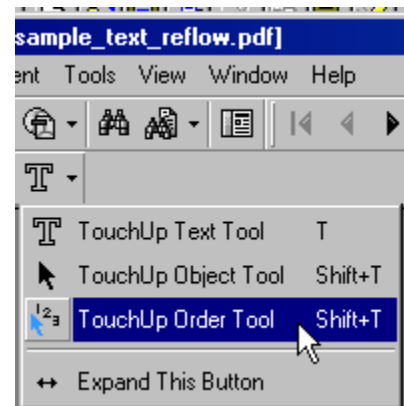
the page designs you've always envisioned. InDesign 1.0 and 1.5 set the stage for this by offering a range of innovations from multiple undo/redo to advanced typographical support. Now InDesign 2.0 redefines creative versatile transparency controls, easy-to-use table features, expanded OpenType support, and new preview controls that reduce on-screen distractions.

Figure 1. Graphics tools

Adobe® InDesign® 2.0

Fortunately, there is a Acrobat tool, the **TouchUp Order Tool**, which is found under the **More Tools** option next to the **TouchUp Text Tool** icon. Unfortunately, it is not the easiest or most intuitive tool to use. Adjusting the text flow order of a document can be tedious, especially for documents that are not visually simple.


3. Click on the **Actual Size** icon to return the document to normal view, then zoom in on the page to at least 150%.
4. Open the **More Tools** menu next to the **TouchUp Text Tool** and select the **TouchUp Order Tool**.



Instructor Notes: Changing the reflow order is not difficult, but it can be tricky. You are advised to go through this exercise yourself a few times before presenting it.

Notice the boxes placed around every frame in the document, each with a number, as shown in the following figure. The number determines the sequence in which the text in each box appears when the text is reflowed. A quick glance will show you why the reflow sequence of the elements on page one does not match the logical sequence. The Title is numbered last (91) on the page, and the two columns of the first paragraph are numbered in discontinuous sequences (31, then jumping to 56), to name a few issues.

85



Adobe

91

Adobe® InDesign®

version 2.0

1 New Feature Highlights

49 ing new
50 dards for
51 essional
52 ut and
53 gn


2 he InDesign 2.0 software sets new
3 dards for professional layout and design.
4 erving the creative tools you've always
5 ted. How do we know? Because you—and
6 gniers and print professionals like you
7 lds—helped us to define what a page
8 ut application should really be. With
9 es on 2.0, you can
11 re your inspiration. 12 sign 2.0
13 ices groundbreaking creative features
14 ransform how you approach page
15 n. For the first time, you can create
16 ble drop shadows, feathering, and
17 parent effects directly in a desktop
18 layout application. You can even place
19 e Photoshop® files with trans-parent
20 grounds and soft edges. In addition,
21 sign 2.0 takes the pain out of creating
22 s, so you can focus on their design.
23 ended OpenType® font support and new
24 ew controls complete the new creative
25 res in InDesign 2.0.
27 master of deadlines. 28 sign is packed
29 ntuitive enhancements that streamline
30 rate design tasks, saving you time and
31 ag you to be more

56 tive. Now, with InDesign 2.0, you can
57 re long documents as easily as you would
58 documents, including generating tables
59 ntents and detailed indexes. The enhanced
60 raph composer (formerly called the multi-
61 mposer) automatically sets optimal line
62 s for each paragraph. An efficient new
63 ng interface delivers precise, reliable
64 s. And performance enhancements ensure
65 ou can work more productively with
66 sign.
68 more efficiently through tight Adobe
69 tion. 70 sign 2.0 integrates better with
71 leading graphics applications—Adobe
72 shop, Illustrator®, and Acrobat®—than
73 ther page layout application. It shares
74 on commands, tools, palettes, and
75 ard shortcuts with these applications, as
76 as core Adobe technologies such as the new
77 e Color Engine included in InDesign 2.0.
78 shop 6.0, Illustrator 9.0/10, and Acrobat
79 n addition, InDesign 2.0 exports Acrobat
80 DF files. The Japanese and Roman versions
81 Design 2.0 can exchange version 2.0 files
82 se of their common file format. InDesign
83 so offers native support of the Mac OS X
84 n 10.1 operating system.

54 pture You
55 piration

33 s you're in the
34 of a creative
35 you want your
36 are to work
37 ively—putting
38 ht tools at
39 fingertips
40 ut getting
41 ur way.
42 e InDesign is
43 ed to give you
44 eative freedom
45 ave, removing
46 ary software
47 rants and
48 ng you produce

32



90 e 1. Graphics tools

86 age designs you've always envisioned. InDesign 1.0 and 1.5 set the stage for this by offering a range of innovations
87 multiple undo/redo to advanced typographical support. Now InDesign 2.0 redefines creative freedom with versatile
88 nergy controls, easy-to-use table features, expanded OpenType support, and new preview controls that reduce on-
89 n distractions.

Fixing The Reflow Order

Now the fun part. The numbering scheme is tricky to change without making a mistake. If you get to an unrecoverable state, click on some other tool, like the Hand Tool, to start over. Use the following advice loosely and try experimentation.

1. **Right-clicking** on one of the numbered boxes will display a reflow command menu. Start the text flow sequence with the Title. Therefore, non-intuitive as it

may sound, right-click any numbered box *except* the Title, and select **Make First**.

2. **Left-click** the Title frame.

Its number should change to 1.

3. **Left-click** each frame on the page in the sequence in which you want it to flow.

The numbers of the frames continue to change in sequence as you click on them.

4. You also can use the options in the right-mouse button menu to modify numbers in the boxes. The **Swap with Next** and **Swap with Previous** commands from this menu are very useful for fixing minor local reordering problems without endangering the order of previous frames. It takes a bit of practice to use the options effectively.
5. If you hold down the control key and left click a number, Acrobat places that number in the *next* box you left-click, and the numbering sequence continues from there. The rest of the numbers shift one place to replace the number you just moved. This seems to be the most effective method for fixing mistakes in sequencing.
6. Using the TouchUp Order Tool is similar to solving a sliding puzzle game; be prepared for surprises. For example, you lose the numbering sequence if you open the right mouse button menu. In that case, the next number box you click returns to number 1. You can fix that by **Ctrl-left-clicking** the number you really wanted there, and then clicking the box that now holds the errant number 1.
7. The page that is most out of sequence is page one. The other two are mostly correct. The TouchUp Order Tool can be frustrating, but do your best to fix the text sequence order without spending an inordinate amount of time on it.
8. Save the file and click the **Reflow** Tool to see the results.

You should see a marked improvement in the sequencing of the text over the original, similar to the screen shot below.
9. Narrow the size of the window and increase the zoom level.

The text should reformat correctly. You may have trouble changing the sequence number for the text caption under the main graphic, probably because it lies fully inside a graphics frame. As a consequence, it will end up out of sequence at the end of the document, as is the case in the screen shot below. The ultimate solution is not to fully embed captions inside graphics frames in the original document

Adobe® InDesign® version 2.0

Setting new standards for professional layout and design

New Feature Highlights

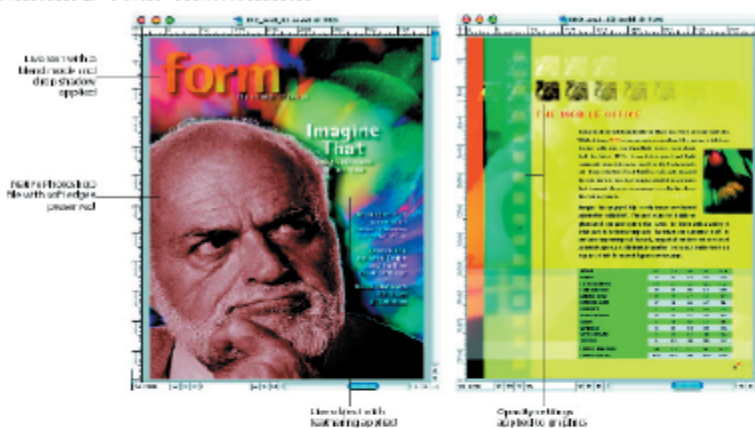
Adobe InDesign 2.0 software sets new standards for professional layout and design, delivering the creative tools you've always wanted to know? Because you—and designers and print professionals like you worldwide—helped us to define what a page layout application is. With InDesign 2.0, you can:

- **Capture your inspiration.** InDesign 2.0 introduces groundbreaking creative features that transform how you approach page design. Now, you can create editable drop shadows, feathering, and transparent effects directly in a desktop page layout application. You can place Adobe Photoshop® files with transparent backgrounds and soft edges. In addition, InDesign 2.0 takes the pain out of creating and focusing on their design. Expanded OpenType® font support and new preview controls complete the new creative features in InDesign 2.0.
- **Be the master of deadlines.** InDesign is packed with intuitive enhancements that streamline elaborate design tasks, saving you time to be more

creative. Now, with InDesign 2.0, you can prepare long documents as easily as you would short documents, including generating table of contents and detailed indexes. The enhanced paragraph composer (formerly called the multi-line composer) automatically sets optimal line breaks and paragraph spacing. An efficient new printing interface delivers precise, reliable results. And performance enhancements ensure that you can work productively with InDesign.

- **Work more efficiently through tight Adobe integration.** InDesign 2.0 integrates better with other leading graphics applications—Adobe Photoshop®, Adobe Illustrator®, and Acrobat®—than any other page layout application. It shares common commands, tools, palettes, and keyboard shortcuts, as well as core Adobe technologies such as the new Adobe Color Engine included in InDesign 2.0, Photoshop 6.0, Illustrator 6.0, and Acrobat 5.0. In addition, InDesign 2.0 exports Acrobat 5.0 PDF files. The Japanese and Roman versions of InDesign 2.0 can exchange files because of their common file format. InDesign 2.0 also offers native support of the Mac OS X version 10.1 operating system.

Capture Your Inspiration



When you're in the flow of a creative idea, you want your software to work intuitively—putting the right tools at your fingertips with your way. Adobe InDesign is designed to give you the creative freedom you crave, removing arbitrary software constraints and helping you create the page designs you've always envisioned. InDesign 1.0 and 1.5 set the stage for this by offering a range of innovations from multiple advanced typographical support. Now InDesign 2.0 redefines creative freedom with versatile transparency controls, easy-to-use table of contents and detailed indexes, and new preview controls that reduce on-screen distractions.

Figure 1. Graphics tools



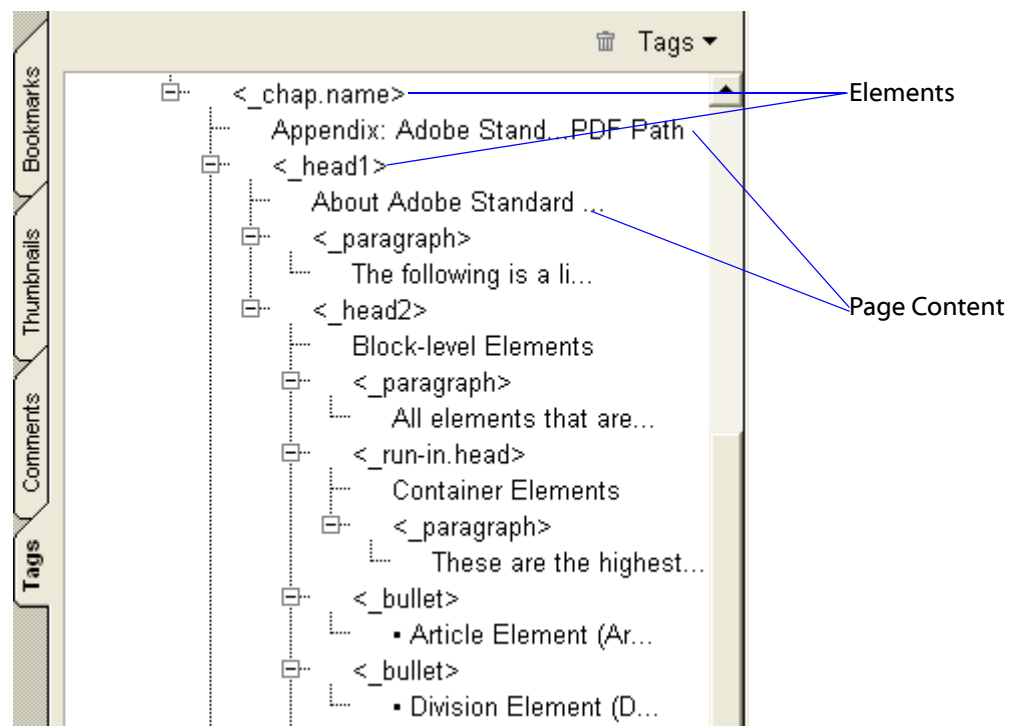


Fixing Content Problems

Instructor Notes: The following paragraphs and screen shots comprise a somewhat lengthy overview of selecting types of elements in a PDF document and creating new PDF Tags for them. In the exercise that follows, students will use the sample file [MktPlanSolution.pdf](#). Have them open that file now, so they can following along with these explanations.

You may find there are situations where the actual document content does not match what has been processed into the PDF tags. This can occur, for example, when a watermark has been picked up and incorporated into the text. There may be other circumstances where text that should be processed specially, like a URL or cross reference, is treated like regular text. You may decide that certain charts or tables require extended explanations to convey content that a sighted audience does not need.

You can use the Tags Palette to create new tag elements to address all of these situations. Adding new elements to the structure tree is a two-step process. First, you must create the appropriate element and insert it into the correct place in the structure tree. Second, you must tie specific page content to the new element. If you neglect to perform this last step, a screen reader ignores the new element.



Study the existing tags in your document to get a feel for what is being used to represent paragraphs, headings, lists, sections, and so on. For consistency, use the same tag types that are currently in your document. If you make a mistake or are not sure what tags to use, do not worry. Screen readers currently do not use the tag type when they process a document. For more information on the types of

tags that are available, refer to the appendix in *Advanced Techniques for Creating Accessible Adobe PDF Files*.

In this section, you will first learn about different tools that you can use to select page content in Acrobat. Once you are familiar with these techniques, you will learn the step-by-step procedure to add new elements to your Tagged PDF document.

Selecting Text

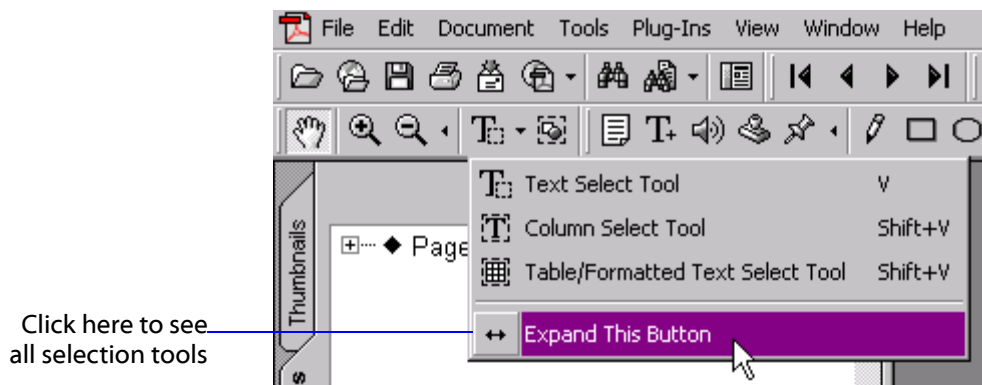
To create a child element from specific content in your PDF document, you must be able to select the desired text. Acrobat provides three tools that you can use to select text: the *Text Select* tool, *Column Select* tool, and the *Touchup Text* tool.

- The Text Select tool is useful for selecting small amounts of text, including discrete words. It is the most convenient tool to use. You can use the Text Select and Column Select tools interchangeably, although there are situations when the Text Select tool works where the Column Select tool does not.
- The Column Select tool only selects text within the bounding box that you sweep with the mouse. It is good for selecting large amounts of text or selecting chunks of text from a multi-column layout. The Column Select tool needs to be used with care. When it works, it is the most convenient of the selection tools. However, if the page content order is not correct, you may think that the Column Select tool has picked up text that it has not really selected. You can detect a problem by slowly selecting text with the column tool. If Acrobat highlights the selected text in an odd way, for example, line 1 highlights, followed by line 3 and then line 2, then the Column Select tool is not working properly. Always verify the text order of selected text by doing a save as **Text (Accessible)**.
- The Touchup Text tool can also be used to select text, although most users do not use it as frequently. Of the three available tools, it is the most reliable but you can only use it to select, at most, one line at a time. Reserve its use for situations where the Text Select tool and Column Select tool are not picking up the text you want.

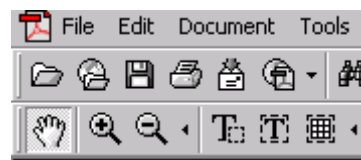
By default, only the Text Select tool is shown in the Acrobat toolbar. In order to expand the selection toolbar, click the down arrow next to the Text Select tool icon




and select **Expand This Button** from the popup menu.



The expanded toolbar is shown below. The table/formatted Text Select tool is used for copying tables in order to paste them into other applications. You cannot use it to tag tables.



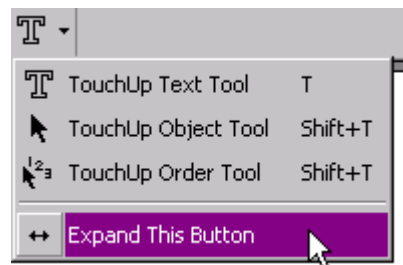
Selecting Graphics

Normally, you use the Touchup Object tool  to edit a graphic in your PDF document. The Touchup Object tool also provides a convenient means of selecting multiple objects that you wish to tag with a Figure element. Using the mouse, you can either left-click a specific object or sweep across the objects you wish to select, holding down the left mouse button. Acrobat highlights all the graphic objects found in the area.



Often, what appears to be a single graphic is a composite of several objects. To be on the safe side, sweep with the mouse to select the object.

By default, the touchup toolbar shows the last selected touchup tool. You can expand this by clicking on the down arrow next to the current touchup tool icon and selecting **Expand This Button** from the popup menu.



Creating a New Child Element

[Instructor Notes: This is a hypothetical example intended as an overview of the procedure. A real example follows in the next exercise.](#)

Once you have selected the text or graphic for which you want to create a PDF Tag in your document, you would then use the following general procedure to create a new Tag element for it in the Tags Palette. Assume for this hypothetical example that there is graphic on one of the pages of the document that lacks a PDF Tag and needs one.

To create a new element:

1. Open the tagged PDF document in Acrobat. Locate the page in the document that contains the graphic for which you want to create a new PDF Tag. Open the Tags Palette.

2. Decide where the new Tag should be placed in the PDF Tag hierarchy:

Select an element in the document which should come just before the graphic in the logical flow. If that is a text element, the Text Select tool would probably be the most appropriate one to use.

Open the Tags drop down menu and select **Find Element From Selection** to find that element's Tag position in the Tags Palette. Refer to ["Finding the Associated Element for Document Content" on page 9](#) for more information on how to find a Tag element associated with particular page content.

3. In some cases, you may find that there is already a PDF Tag associated with a particular document content, but that it is not the type you want, or that it is not appropriate for your purposes. If that is the case, delete the Tag element by selecting it in the structure tree and pressing the Delete key.

In general, the Tags you are interested in, and the ones that are critical for accessibility purposes, contain either text or graphics with alternative text. A number of other Tags may have been generated in the process of creating the PDF file which are superfluous for accessibility purposes. For the most part, these can be removed from the Tags Palette if you wish to do so. For example, there is a <Shape> Tag just above the <TextBox> Tags you worked with in the exercise [Fixing Logical Read Order Problems](#), which has several PDF_Path child elements. If you turn on Associated Content Highlighting, you will see that these Tags refer to the graphic frames around the text boxes, and that the <Shape> Tag itself refers to a group of three of these boxes. These Tags lend nothing to the accessibility of the document, and, unless you wish to add alternate text to them, can be deleted without affecting accessibility.

4. Find the Tag that is the parent for the new Tag you wish to create. The parent Tag may be considerably further up in the Tag hierarchy from the position of the new Tag. Typically, the parent Tag for the new Tag will be the same as for the adjacent element you selected in Step 2, above.

Click the element that will be your new element's parent Tag.

5. Open the Tags drop down menu, or select the parent Tag and right-click on it to bring up the Tags menu. Select **New Child Element**.
6. From the popup menu that appears, select the Tag element type you wish to insert.

For more information on the types of elements that are available, refer to the appendix in *Advanced Techniques for Creating Accessible Adobe PDF Files*.

The new Tag element becomes the *last* child under the parent Tag. If you do not see the new Tag there, you may have inadvertently placed it as a child element under the wrong parent.

7. Move the new Tag to the location in the Tags hierarchy you identified in Step 2, above.

The new Tag you just created is a placeholder. You must now tie the element to the actual page content.

8. Click the new Tag to select it. This is important, because the selected Tag will become the parent for the document content you are about to specify.
9. Using the appropriate selection tool, select the desired text or graphic. For this hypothetical example, you would use the Touchup Object tool.
10. From the **Tags menu**, select **Create Child Element From Selection**.
11. **Save** the PDF file.

Modifying the Tag Type of an Element

There may be situations where page content has been associated with the wrong element type. When this happens, you do not need to delete and recreate the element. You can simply modify the element's type.

To modify the type of an element:

1. Open the tagged PDF document in Acrobat, bring up the Tags palette, and locate the element that you need to change.
2. Click the element that you want to modify.
3. Open the Tags menu and select **Element Properties**.
4. In the **Type** field in the **Element Properties** dialog box, delete the existing Tag element type name and enter a new one.

For more information on the types of elements that are available, refer to the appendix in *Advanced Techniques for Creating Accessible Adobe PDF Files*.

5. **Save** your PDF file.



Incremental changes to a PDF file are added to the end of the file. To compress your PDF file, and make it more efficient, you should do a Save As (as PDF) to the same file name that you have been using.

Use your discretion to decide whether a graphic should be accessible. Graphics that are not important for page content are also referred to as *artifacts*. These can be ignored entirely if they do not have an associated PDF Tag. If you want a graphic to be accessible, always create a **Figure** Tag for it. After you have created a Figure Tag, make sure you edit the Tag's properties and enter alternate text for it.



Exercise: Creating New Tag Elements

In order to do this exercise, you must have Acrobat 5.05 installed on your machine. You must also have downloaded and unzipped the file **TagsPalette.zip** from the Accessibility training Web site.

In this exercise you will work with a partially Tagged PDF file. The file was originally created from an application that could not create Tagged PDF. You will practice tagging both text and graphic content. You can find the solution for this exercise in **MktPlanSolution.pdf**.

At the end of the exercise you will be able to:

- Create heading, paragraph, and figure elements
- Use the Column Select tool and Text Select tool to select text
- Use the Touchup Object tool to select graphics
- Add alternative text to a Figure element by editing the element's properties
- Verify that elements that you created are accessible

Tagging Text

1. Open **MktPlan.pdf** in Acrobat.
2. Open the Tags palette, expand the structure tree, and look over the current state of the document.

Use **Turn On Associated Content Highlighting** from the Tags menu to get a feel for how much of the document has been tagged.

On page 1, most of the content has been tagged except for the last heading and paragraph. You will add a Heading 2 and Paragraph element to the structure tree so this content will be accessible.

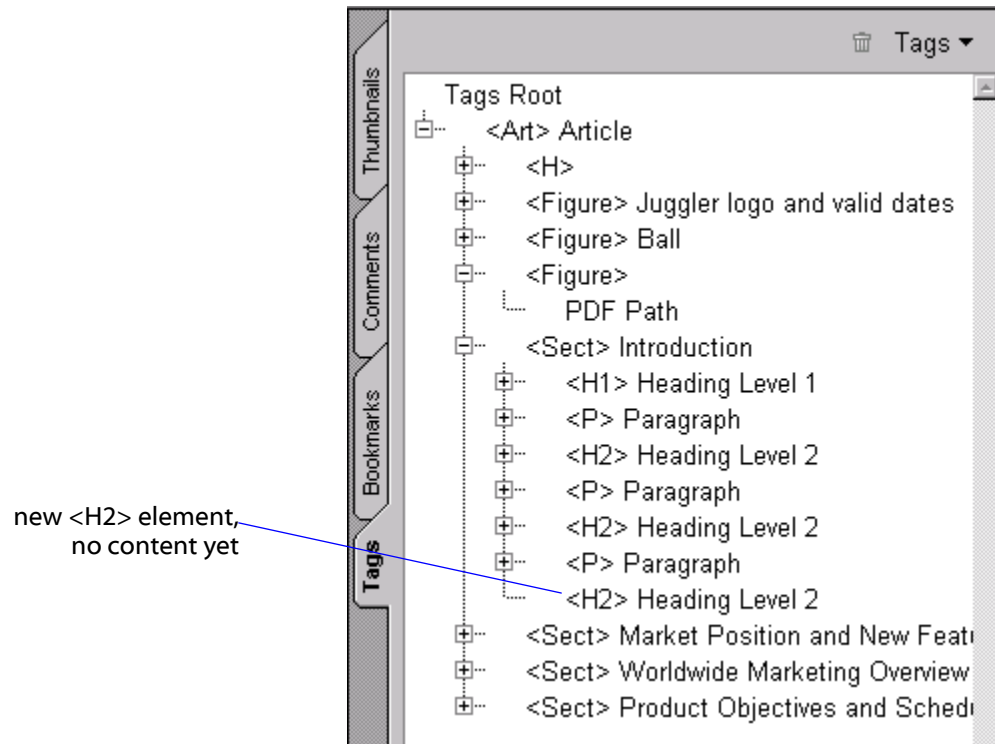
3. In the Tags palette, find **<Sect> Introduction**.


This is the parent element.

4. Click **<Sect> Introduction** to highlight it.
5. Click on the **Tags** drop down menu and select **New Child Element > Heading Level 2 Element**.

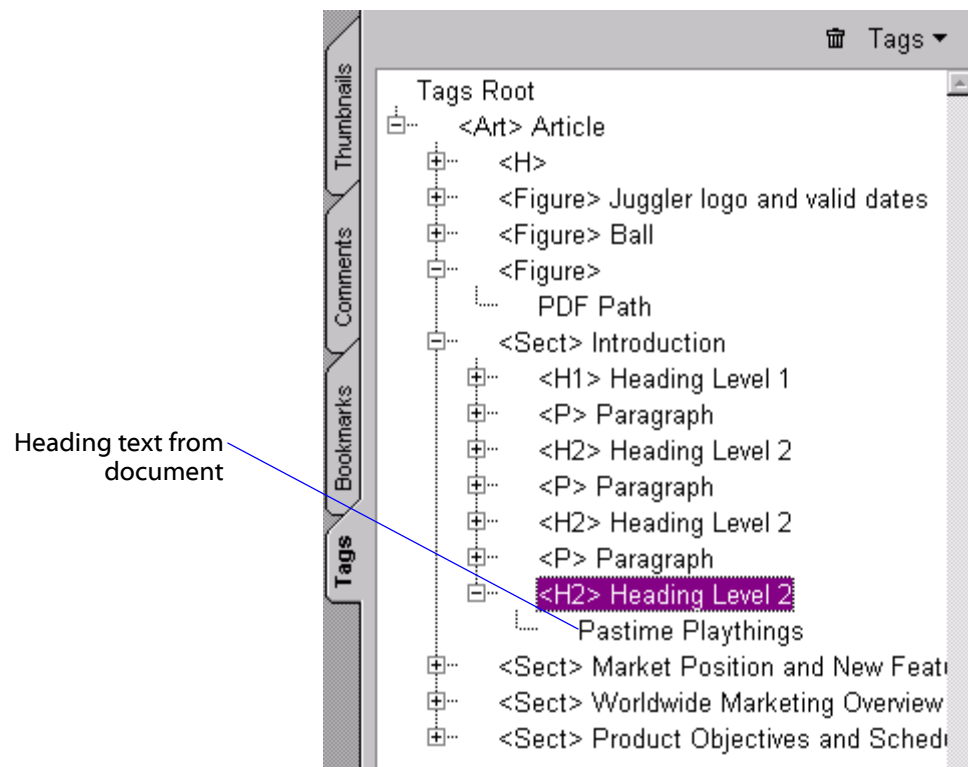
Acrobat will insert an empty **<H2>** element as the last child element under **<Sect> Introduction**.

Instructor Notes: When a New Child Element is created, it is always placed as the last child in the hierarchy under the tag element that was selected at the time the new child was created. This may be a little confusing at first. Often, the newly created tag will then have to be moved into the correct position in the logical order.



6. Click the new **<H2>** element to select it.
7. In the Acrobat toolbar, click the **Column Select** tool .
8. Move the mouse cursor over to the document pane.
Notice that the cursor is now an I-bar with a bounding box around it.
9. Hold down the left mouse button as you select the text heading **Pastime Playthings**.
10. Release the mouse button to complete the selection.
11. Click on the **Tags** drop down menu and select **Create Child Element From Selection**.

The structure tree should appear as below.




12. Now add a Tag for the paragraph below the Heading:

Click **<Sect> Introduction**.

13. **Right-click** on the **<Sect>** Tag to bring up the Tags menu and select **New Child Element > Paragraph Element**.

Acrobat will insert an empty **<P>** element as the last child element under **<Sect> Introduction**.

14. Click the new **<P>** element to select it.

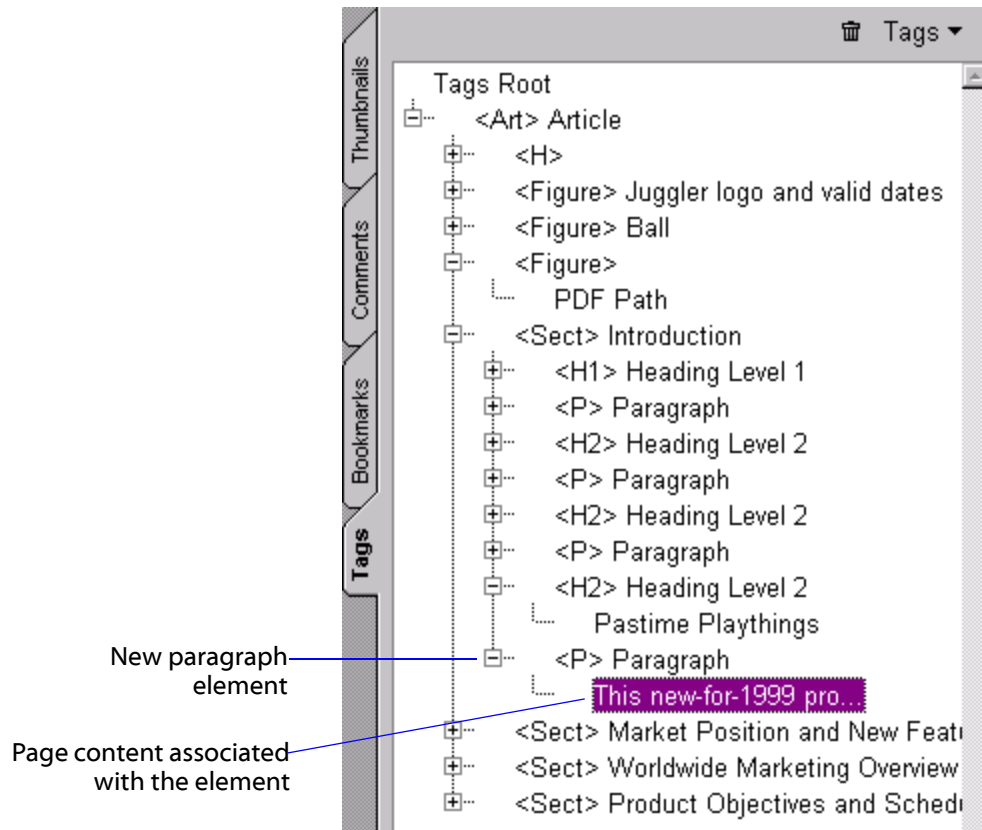
15. In the Acrobat toolbar, click the Text Select tool .

16. Move the mouse into the document pane and select the last paragraph on page 1.

17. Open the Tags menu and select **Create Child Element From Selection**.

18. Save your changes.

The structure tree should appear as below.



19. To test your changes, turn on content association and step through the elements under **<Sect> Introduction** to see if the content is in the right order.

20. **Save** the file as **Text (Accessible)**.

21. Open the text file with an editor and make sure the heading and associated paragraph for Pastime Playthings now appears at the end and in their entirety.

If the text does not appear in the file, you may not have created a child element based on the text in the document.

Note that if the new Heading and paragraph had not been the last elements on the first page, you would also have had to move the Tags to their correct position under the **<Sect> Introduction** Tag


Tagging Graphics

Instructor Notes: Students will not be tagging everything in the sample document, so there will be gaps in tag structure. The pie chart on page 3, for example, is not tagged

1. Open **MktPlan.pdf** in Acrobat.

On page 1, there is a graphic of a green car at the bottom of the page. You will create a figure element for this car.

2. In the Tags palette, find **<Sect> Introduction**.

3. Create a new **<Figure>** element under **<Sect> Introduction**:
Right-click on **<Sect> Introduction** and select **New Child Element > Figure Element**.
4. Select the **Touchup Object** tool . The green car is a single object. **Left-click** it to select it.
5. **Right-click** on the **<Figure>** Tag you just created, and select **Create Child Element From Selection**.
6. Add **Alternate Text** to the graphic by right-clicking the **<Figure>** element and selecting **Element Properties** from the popup menu.
7. In the **Alternate Text** field in the **Element Properties** dialog, enter the text *Sample vintage vehicle*.
8. At the bottom of page 3, there is a bar chart labelled "Worldwide Projected Revenue Goals For 1999 - 2001" showing differences in revenue projections for Jugler Toys versus its competitors for the years 1999 through 2001.

Add **Alternate Text** for the barchart figure, describing the percentages for each company.
9. Test your changes by saving the file as **Text (Accessible)**.

Acrobat saves the alternate text for the figure in the file. If you had a **<Figure>** element with no alternate text, it would be invisible to the screen reader. If you wish, you can use a screen reader to process the PDF file.

Extra Credit

[Instructor Notes: Use this exercise if you have time, or give it to the fast ones in the group. Step by step instructions are included for creating a tag structure for Lists, but students are on their own in regard to creating the table tag structure.](#)

MktPlan.pdf also contains a bulleted list and a table on page 4. The Acrobat Distiller will correctly convert lists and tables from the original authoring tool, for example, MS Word, InDesign, or FrameMaker, to Tagged PDF. **MktPlan.pdf**, however, was not originally created as a Tagged PDF document. As a consequence there are no Tags in the Tags Palette for those content elements and they will not be accessible by a screen reader.

In Tagged PDF, lists and tables are elements represented by the Tags, **<L>** and **<Table>**. They themselves are comprised of child elements **** and **<TR>**, which in turn are comprised of other child elements, **<LBody>** and **<TH>** or **<TD>**. Most screen readers recognize the tags that comprise lists and tables, and note to the user that a list or table is being read.

You may recognize the similarity of these tags to HTML. Each item in a list, **<L>**, should have its own list item, ****. Since an item in a list could potentially be comprised of several paragraphs, each paragraph should be tagged with a list body, or **<LBody>** element. The **<Table>** element should contain a **<TR>** element

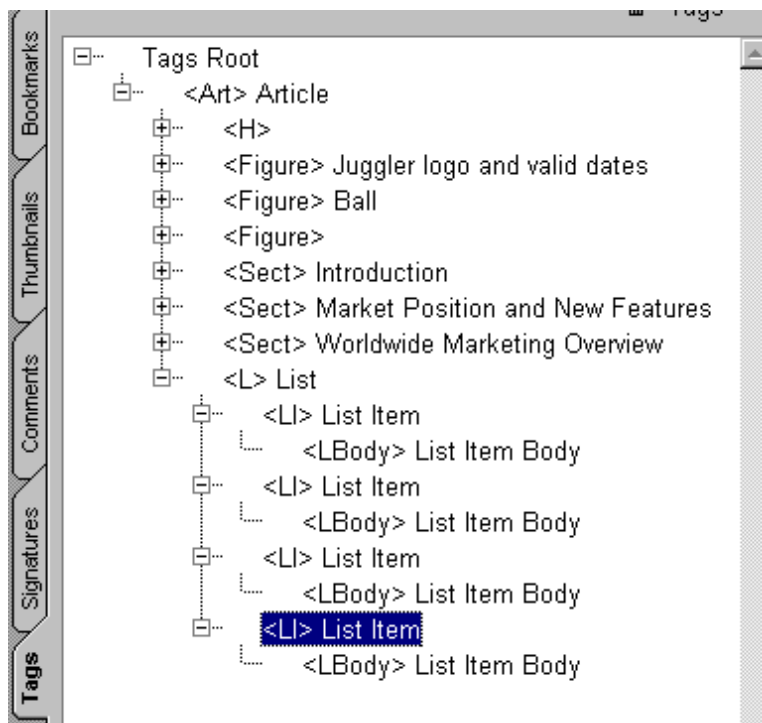
for each row, including the heading. Within the **<TR>** element, there should be a **<TD>** (data cell) or **<TH>** (heading cell) element for each cell in that row.

In this supplemental exercise, use the following two step procedure to add tag structure to the list and table on page 4:

1. Determine what elements are needed in the Tag structure to represent the content of the list and table, and add those tag elements to the Tags Palette.
2. Select the content elements in the document that correspond to the tag elements in the Tags Palette, and add them as child elements of their respective Tags.

Start by analyzing the list on page 4. It needs a **List** tag, and four **List Item** tags each containing a **List Item Body** tag. The List element is going to be a child element of the **<Art>** Article tag. It should be placed at the bottom of the hierarchy, which will happen anyway, by default, because the last tagged content in the document is the graph on page 3.

1. **Right click** on the **<Art>** Article tag and select **New Child Element > List Element**. The **<L>** List tag should appear below the last **<Sect>** tag and at the same level in the tag structure.
2. **Right click** on the **<L>** List tag and select **New Child Element > List Item Element**. The **** List Item tag should appear below the **<L>** List tag and one level deeper in the tag structure.
3. **Right click** on the **** List Item tag and select **New Child Element > List Item Body Element**. The **<LBody>** List Item Body tag should appear below the **** tag and one level deeper.
4. **Repeat** steps 1 through 3 for the other three bullet items in the list on page 3. When finished, the new Tag hierarchy should look like this:



Instructor Notes: The trick in getting the placement right in the tag hierarchy is to always right click on the parent tag, or select it, before selecting New Child Element: the new element is always created as the child of the selected tag. If students make a mistake in the placement of the tags, they can move them to the correct position or delete them and try again.

Next, you need to specify what actual content elements in the document are represented by the new tags to the Tags Palette. In the Tags Palette, the last, or most deeply nested tag in a parent-child structure is the one that is associated with the content in the document. So in this case, the <LBody> List Item Body tags are the ones you are going to connect to each of the bulleted items in the list on page 4.

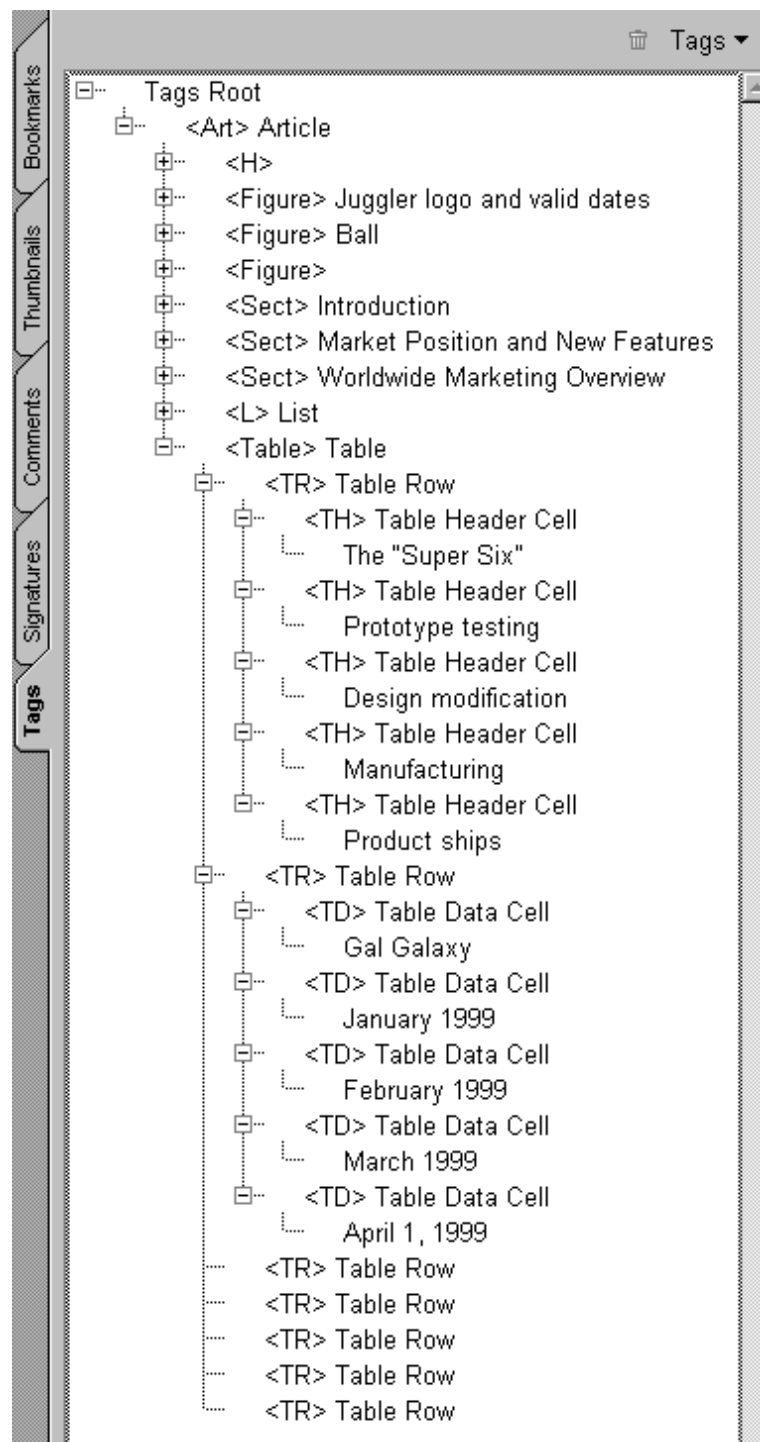
5. Go to the list on page 4. You can use either the **Text Select** tool or **Column Select Tool** to select the entire text of the first bulleted item.
6. **Right click** on the first **<LBody> List Item Body** tag in the List structure and select **Create Child Element From Selection**. If you click on the plus sign that now appears to the left of the <LBody> tag, you will see that the text from the first bulleted item in the list on page 4 has been added to the tag structure as a child of the first <LBody> tag.
7. **Repeat** Steps 5 and 6 for the other three bulleted items in the list.
8. **Save** the file as **Text (Accessible)** and check to see that the list items are now included. Alternately, use a screen reader to see if the list items are read, and secondarily, to see if the bulleted items are identified by the screen reader as items in a list.

Now you are on your own. Using the procedures you learned above for creating PDF Tag structure for the list on page 4, create a PDF Tag structure for the table at the bottom of page 4.

- Create the tag structure for the table in the Tags Palette.
- Select the text in the table's header and body cells and associate it with the appropriate tags in the Tags Palette. Selecting the text in the table cells can sometimes be a bit tricky. Use the Touch Up Text Tool, which seems to work best in terms of being the most controllable when selecting text in just one particular cell.

Even using the Touch Up Text Tool, you may have trouble selecting the "Manufacturing" cell heading because it actually exists in its own text box behind the text box that contains the other cell headings. A trick is to skip the "Manufacturing" heading and go on to select "Product ships" and associate that with the last <TH> tag. Then, with "Product ships" still selected, press the right arrow key twice, which will cause the cursor to jump back to the "Manufacturing" text box, from which you can select it and associate it with the correct <TH> tag.

- When complete, the <Table> tag hierarchy should look like the following screen shot, in which the first header row and the first data row tags have been expanded:





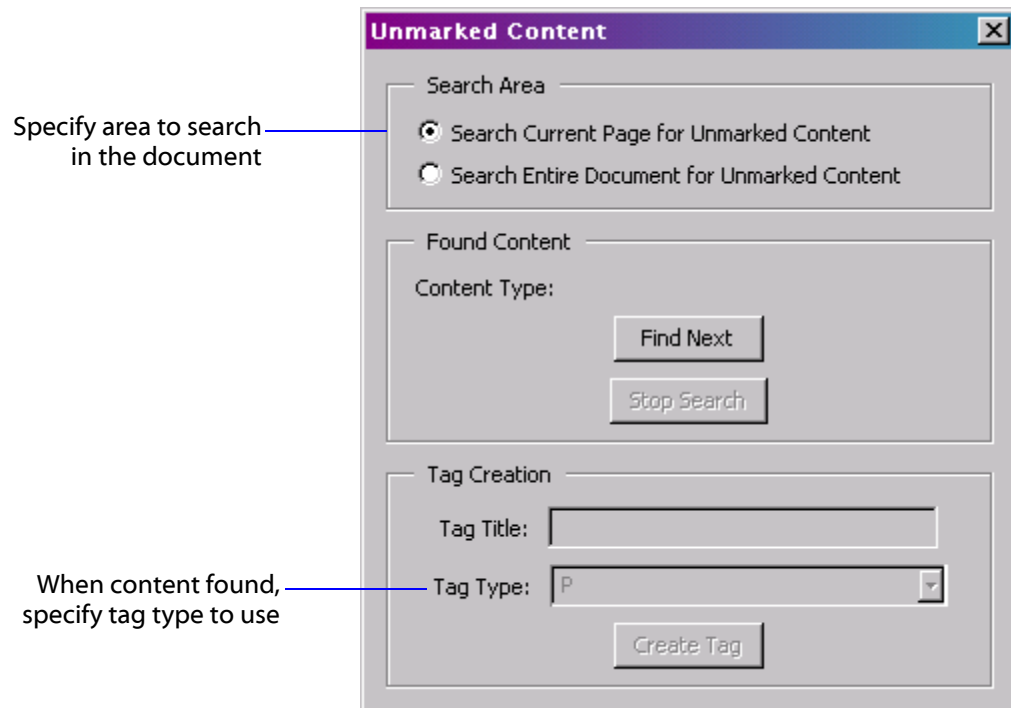
Converting Missing Content

Occasionally, the conversion process to Tagged PDF neglects to create corresponding tag elements for some of the page content. If page content has no associated tag element, it will not be accessible to a visually-impaired person. It is not necessary to convert every single piece of text and graphic. Some text, like page headers and footers, may not provide any significant information to a visually-impaired person. Sometimes graphics are used to add color and visual appeal to a document. These document elements, which are referred to as *artifacts*, do not need corresponding tag elements if they add nothing to the intended meaning of the document.

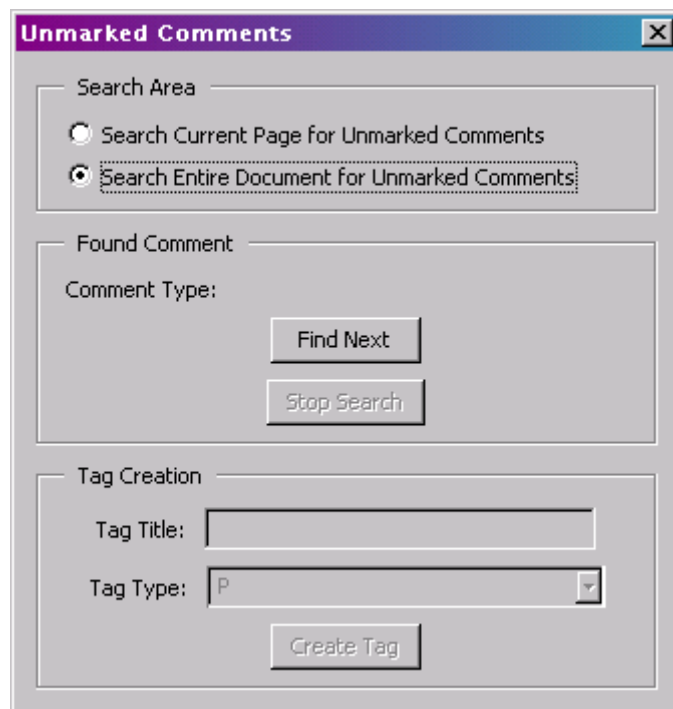
Although the Accessibility Checker can locate untagged page components, you can also use two Tags palette commands to dynamically create a tag element for document content that is not in the structure tree. The commands are **Find Unmarked Content** and **Find Unmarked Comments**. Both of these commands allow you to create a tag of your own specification for the found object. You can also elect to skip creating a tag. When Acrobat finds an object that has no associated tag, it highlights the object with a border. You can use the document pane to examine the object and decide whether you want to tag it. If you do tag it, you need to specify the type of tag to use and, optionally, a tag title. The tag title allows you to easily figure out to what the tag corresponds without having to use content association.

Acrobat automatically inserts a dynamically created tag as the last child element of the currently selected tag. Therefore, to avoid moving tags around, you should determine which content is missing tags and where the surrounding tags are located in the structure tree. Once you know where the missing content's element should go, select the parent tag in the palette pane and then use one of the **Find...** commands to search just the current page.

Use **Find Unmarked Content** to search for text, lists, tables, and so on that were not converted. If you have composite graphic objects that were not tagged, attempts to create a tag for each individual graphic object. You have some options for how to approach composite graphics. You may choose to use the **Find Unmarked Content** command in spite of the fact that it will create unnecessary tags for individual graphic objects, since the hard work is in finding and selecting the correct content to begin with. You can then move the content to the right place in the Tags hierarchy and delete the empty tags. You can choose to rearrange the structure tree to organize the individual objects, or you may decide to use manual tagging for composite graphics.



Due to PDF implementation details, links, such as cross references within a document, and form fields are considered to be comments or *annotations*. Use **Find Unmarked Comments** to search for links, annotations, or form fields that are not present in the structure tree.



Be aware that link elements differ from annotation and form field elements in that links have page content counterparts. Consequently, using **Find Unmarked Comments** to locate untagged links only accomplishes half the job. You must still manually tie the new link element back to the appropriate page content to make it accessible to a screen reader. If the link is embedded in a paragraph, you also have to manually split the paragraph into two separate paragraph tags, one coming before the text of the link, and one following the text of the link.

You need to use a screen reader to verify that a link is accessible. A screen reader uses audio cues to let a user know that the text it is reading is a link that can take the user to a different part of the document or to a Web site. When Acrobat saves a file as **Text (Accessible)**, links are not treated any differently than regular text, so using Text (Accessible) is not a useful way to test whether links are active in the logical structure.



Exercise: Converting Missing Content to Tags

In order to do this exercise, you must have Acrobat 5.05 installed on your machine. You must also have downloaded and unzipped the file **TagsPalette.zip** from the Accessibility training Web site.

In this exercise you will work with a Tagged PDF file with cross references that are not distinct tag elements. You can find the solution for this exercise in **UserGuideSolution.pdf**.

Finding Unmarked Comments

Instructor Notes: Reinforce making a copy of the exercise file in Step one, otherwise someone won't do it. They will need a copy of the original in order to compare the original with the modified version.

1. Make a copy of **UserGuideStart.pdf**. You will need to refer to the *unmodified file later*.
2. In Acrobat, open the copy of **UserGuideStart.pdf**.
3. Open the Tags palette.
4. First, locate the untagged cross references and figure out where they should belong in the structure tree.

Select **Tags > Find Unmarked Comments**. Select **Search Entire Document for Unmarked Comments**.

5. When Acrobat finds an unmarked comment, the **Tag Title** and **Tag Type** fields will become available for selection. You may need to scroll in the document pane to find the highlighted link. Make note of the page number and general location of the cross references, but do not create tags for them yet. There are only two unmarked links, on page 9, *Update Student Information* and the numeral 5 in *page 5*.
6. Use the **Text Select Tool** to select any part of the line containing the link *Update Student Information*. Then use **Tags > Find Element From Selection** to determine approximately where in the Tag hierarchy the cross reference links should be inserted.
7. In the structure tree, select the element that will be the parent of the cross reference links.

Hint: It should be an **<LBody>** element that corresponds to the text for the third numbered list item on page 9.

8. Once more, select **Tags > Find Unmarked Comments**. You may have to position the cursor at the top of the document before using the Find command again. This time you will create the link elements.
9. Enter the following information into the **Unmarked Comments** dialog

- **Tag Title:** *Update Student Info*
- **Tag Type:** Open the drop down menu and select **Link**.

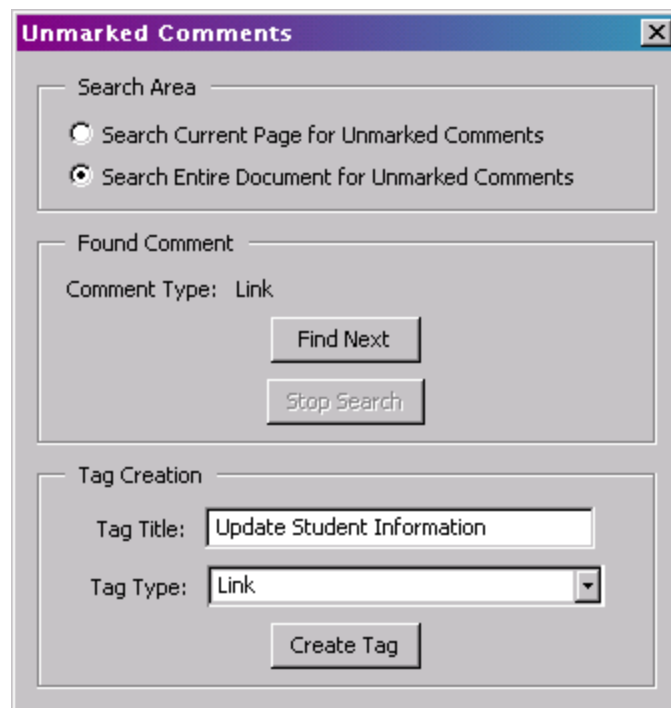
10. Click **Create Tag**.

11. Click **Find Next** to continue your search.

By default, the tag title and tag type are not cleared when Acrobat finds the next unmarked link.

When Acrobat finds the second untagged link, enter the following information:

- **Tag Title:** 5
- **Tag Type:** Open the drop down menu and select **Link**.



12. Save the file as **Text (Accessible)**.

13. Save the original `UserGuideStart.pdf` as **Text (Accessible)**.

Do the files look different? You should notice that the files are identical. If you use a screen reader to process the PDF file you just edited, it does not treat the links any differently than the other text in the document. What is going on?

You may remember that links have to be tied to specific page content. This is the missing piece of the puzzle. In addition, you probably did not modify the original paragraph that included the two links to separate the regular text from the links.

Fixing Missing Comments

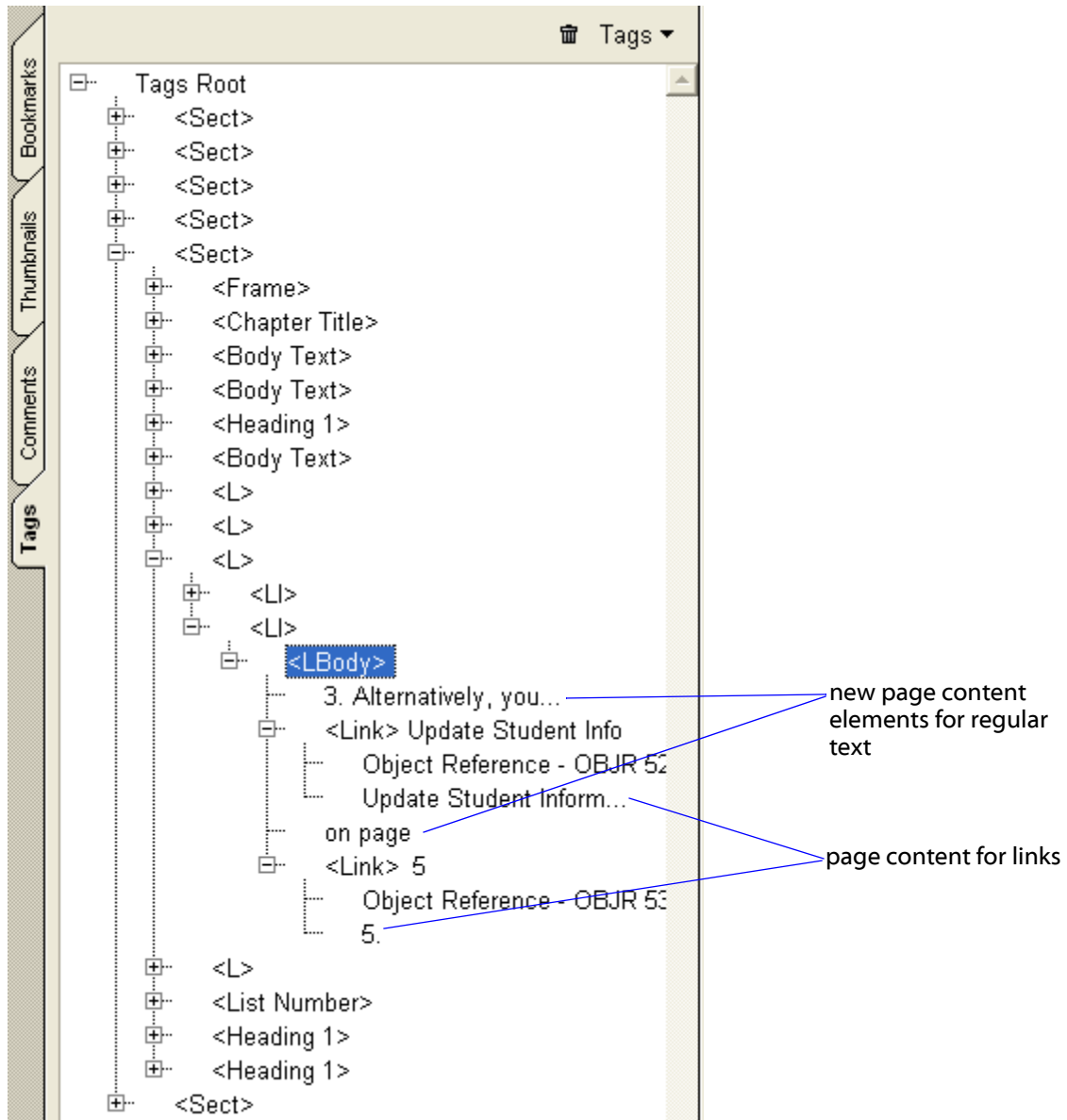
In this step, you will create a compound child structure for the <LBody> tag of the paragraph that contains the two links. You will create a child element for the first part of the paragraph up to the first link and for the part of the paragraph between the two links. You will add the text of the links as child elements under their respective <Link> tags, and then arrange them in the correct order under the <LBody> tag for the paragraph.

1. Use one of the methods you've learned to locate the **<LBody>** parent tag of the two Link elements you just added to the Tags Palette. Click on that **<LBody>** tag to select it.
2. Using the **Text Select Tool**, highlight all the text in list item three, including the list number "3.", and everything else that comes before "Update Student Information".
3. **Right click** on the **<LBody>** tag and select **Create Child Element From Selection**.
4. **Move** the new child element into the correct place, just below the <LBody> tag.
5. Use the **Text Select Tool** to select the text "on page" between the two links, then **repeat** Step 3.
6. Move the new child element into place between the two links.
7. Notice that the text of the links now exist as separate child elements of the **<LBody>** tag. You want them to be child elements of the **<Link>** tags, however. **Move** them into the correct place using the mouse.

Alternately, you could select the text for each link in the document pane, then select the appropriate **<Link>** tag in the Tags Palette, and then select **Create Child Item From Selection**. However, simply moving the tags is easier. You may want to delete any empty tags that appear in the modified paragraph structure, although empty tags are harmless.

Using whatever method you choose, rearrange the Tag structure until it looks like the one shown in the screen shot below.

8. Test your changes using a screen reader. The links should be correctly identified.





Summary

In this module, you learned how to modify the Tag structure characteristics of a PDF file for accessibility purposes. You were introduced to a number of Acrobat commands and utilities available for the post processing modification of a PDF document's logical structure. You were reminded that it is always preferable to control the PDF structure in the original authoring tool, prior to generating the PDF file, if possible. This is because modifications to the PDF Tags Palette cannot be saved and reapplied if the PDF document is re-generated from the original source. Not all authoring tools generate structured PDF however, in which case the Acrobat tools you have been introduced to in this module will be essential for developing an accessible PDF document.

The topics covered in this module were are follows:

Logical Read Order and Page Content Order

Logical read order, logical structure, and logical flow refer to the order in which contents of a document will be accessed by devices like screen readers. The logical flow is determined by the hierarchical structure of the tags in the Acrobat Tags Palette. The logical flow should conform to the sequence in which the document was intended to be read. Altering the logical flow by altering the structure of the Tags Palette does not affect the physical appearance of the document.

Page content order, or text reflow order, refers to the sequence in which the contents of a PDF document are presented when they are reflowed into different display formats. A display format, in this case, might be a handheld PDA. More relevant to accessibility issues, the format might be greatly magnified text within a narrow viewing window, used by someone with low visual acuity. In both cases, the reflow order determines the sequence in which the document content elements are displayed, and the reflow feature allows the contents of the document to be correctly formatted and wrapped to the dimensions of the display window. Reflow order is not necessarily the same as the visual order of elements in the document, nor is it necessarily the same as the logical order of the document. Both logical order and reflow order should be attended to when creating accessible documents.

Fixing Content Problems in the Tags Palette

You learned how to add structure elements to the PDF Tags Palette using the commands in the Tags menu. You also learned to use various Acrobat Selection tools for selecting text and graphics in a document, and how to use Tags menu command to associated document content with new and existing tags in the Tags Palette. You learned how to move elements in the Tags Palette into their appropriate places.

Finding and Converting Missing Content

You learned how to use the Find Unmarked Content and Find Unmarked Comments commands from the Tags menu to find text, tables, graphics, and

cross reference links that were not tagged. You used the Find Unmarked Comments command to find two untagged cross reference links in the document, and you used various previously acquired skills to add a functioning tag structure for the links in the Tags Palette.