



XDOC DATA FORMAT

Technical Specification

Version 4.0—May 1999

Copyright Copyright © 1995–2000 by ScanSoft, Inc. All rights reserved. No part of this publication may be transmitted, transcribed, reproduced, stored in any retrieval system or translated into any language or computer language in any form or by any means, mechanical, electronic, magnetic, optical, chemical, manual, or otherwise, without the prior written consent of ScanSoft, Inc., 9 Centennial Drive, Peabody, Massachusetts 01960. Printed in the United States of America.

The software described in this book is furnished under license and may be used or copied only in accordance with the terms of such license.

Disclaimer ScanSoft, Inc. provides this publication “as is” without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability or fitness for a particular purpose. Some states or jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you. ScanSoft reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of ScanSoft to notify any person of such revision or changes.

Credits *TextBridge* is a registered trademark and *ScanWorX* is a trademark of ScanSoft, Inc.. Other terms used in this manual may be the trademarks of other manufacturers.

Writers: Daniel S. Connelly
Beth Paddock
Rebecca Harvey

© **ScanSoft Inc.**
9 Centennial Drive
Peabody, Massachusetts 01960
Main: 978-977-2000
SDK Technical Support: 978-977-2174
Email: api_support@scansoft.com

XDOC Data Format: Technical Specification

Part Number 00-08137-00

May 1999

Table of Contents

Table of Contents	iii
Preface	v
About This Document.....	v
Documentation Conventions.....	vi
Related Publications.....	vi
Customer Support.....	vi
Overview	1
XDOC Text Markups	1
2.1 Modifiers and Operands.....	2
2.2 Transactions.....	3
2.2.1 Text line transactions	4
2.2.2 Page transactions.....	6
2.2.3 Document transactions	7
2.3 Text Transaction Syntax.....	7
XDOC File Structure	1
3.1 Page Sequence.....	1
3.2 Text Data.....	2
Page Image Analysis	1
4.1 Units of Measure and Coordinate Transformations.....	1
4.2 Page Segmentation.....	3
4.2.1 Topology	5
4.2.2 Lexical order	7
4.2.3 Image zones	7
4.3 Rulings.....	7
4.4 Algebraic Transformations.....	9
4.4.1 Convert page coordinates to image coordinates.....	9

4.4.2 Convert skew information to degrees	11
4.4.3 Convert original image coordinates to deskewed coordinates.....	11
4.5 Fonts.....	13
Transformations	1
A.1 Text Lines.....	1
A.2 Zones.....	3
A.3 Rulings.....	4
Sample XDOC Text	1
B.1 Sample Page Image	1
B.2 Sample XDOC Text.....	2
Character Set	1
Index	1

Preface

This *Technical Specification* describes the ScanSoft Inc. **XDOC** output format. XDOC provides richness of information that can be converted to another output format. It captures the structure of documents as processed by ScanSoft's document recognition products, including **ScanSoft SDK** for Windows platforms.

XDOC data format is one of several text output format options available from the ScanSoft SDK, and other Scansoft products. You can write applications to interpret XDOC Text for use with a word processor, desktop publishing system, or any other applications requiring input from paper or image files.

About This Document

The *Technical Specification* describes the structure and syntax of the XDOC format. The information is aimed at systems developers and applications programmers who are familiar with character formats and markups, and image file formats, as well as their particular platform or OS environment. The specification is organized as follows:


- Chapter 1, "Overview," describes the hierarchy and relationships of the XDOC data format.
- Chapter 2, "XDOC Text Markups," describes the structure and syntax of XDOC text markups.
- Chapter 3, "XDOC File Structure," provides information about the structure of and page sequence in an XDOC Text file.
- Chapter 4, "Page Image Analysis," discusses how ScanSoft's document recognition software initially perceives page image layout, divides it into meaningful areas, and identifies fonts.
- Appendix A, "Transformations," further describes the transformations described in Chapter 4.
- Appendix B, "Sample XDOC Text," consists of a page image and its corresponding XDOC Text file.

Documentation Conventions

Throughout this *Technical Specification*, “you” refers to the application programmer or system developer, and “the user” refers to an application end user.

As described in Table P-1, the *Technical Specification* uses certain graphical elements and formatting to emphasize information and denote meaning in the text.

Table P-1. Documentation Conventions

Convention	Description
bold	Introduces a new term, or the first use of an important term in a chapter; highlights function name.
<i>italic</i>	Denotes titles of manuals or books. Also used to denote generic representations of entries in examples.
monospace	Denotes code examples and file names.
“ ” (quotes)	Denotes titles of chapters and sections in this manual.
	Introduces tips that provide useful information about a procedural step or system function.
Note	Introduces important information about the current subject.

Related Publications

Consult the *Overview* for the ScanSoft SDK for detailed information about writing document recognition applications. Refer to the appropriate user’s guide and installation guide for any other ScanSoft product that produces XDOC format

Customer Support

If you should experience problems working with XDOC that you cannot resolve, contact Technical Support via electronic mail at

`api_support@scansoft.com`

If you prefer to use the telephone, you can call Customer Support at

978-977-2174

Be ready to provide:

- Your software registration number.
- A detailed description of the steps that led to the difficulty.

Overview

XDOC captures the content and structure of compound documents—containing both text and graphics—as recognized and output by ScanSoft Inc. document recognition software. In addition to recognized text, XDOC provides detailed information about the following:

- page layout (X,Y coordinates of text and graphics on the page)
- character formatting (bold, italic, subscript, superscript, underline)
- font metrics
- graphic images
- horizontal and vertical rulings
- document structuring conventions
- lexical classes
- character and word confidence
- word bounding boxes

XDOC provides a flexible grammar that your applications can easily interpret and convert to output formats for use with a word processor or desktop publishing system.

ScanSoft's document recognition software stores XDOC data format information about each document in a set of related files. A single **XDOC Text** file contains information about the text and references any files that describe the graphics. XDOC Text is a ScanSoft-defined document storage language.

In most cases, each graphic element in a document is a separate file written in an image file format, such as **TIFF**.

Figure 1-1 illustrates the hierarchy of the XDOC data format.

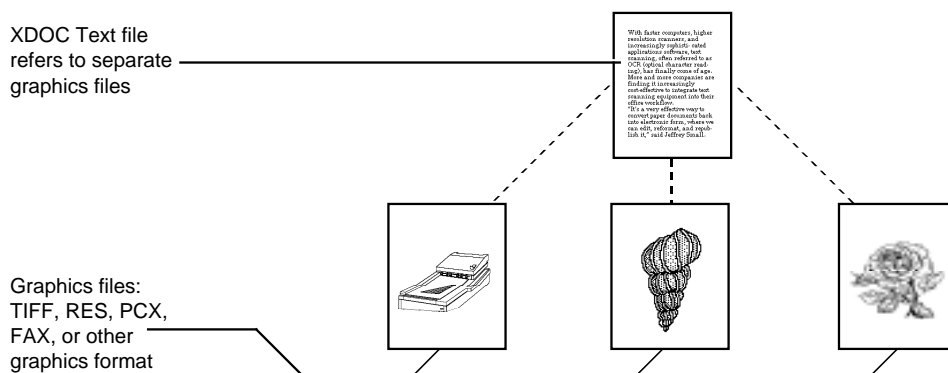


Figure 1-1. XDOC data hierarchy

You can then use these XDOC Text and graphics files as input for your applications (and selected ScanSoft products), to interpret and convert to a format useful to an end-user.

☞ You can also read a graphics file with generic, off-the-shelf software.

The format, structure, and layout of XDOC Text are discussed in more detail in the remainder of this specification.

Chapter 2, “XDOC Text Markups,” details the structure and syntax of the text markups that describe the physical and logical attributes of text in an XDOC Text file.

Chapter 3, “XDOC File Structure,” describes the structure and page sequence of an XDOC Text file. This information can help you determine the best way to read and interpret an XDOC Text file with your applications.

Chapter 4, “Page Image Analysis,” explains how the document recognition software initially perceives the layout of a page image, divides it into meaningful areas of graphics or text, handles rulings, and identifies fonts.

XDOC Text Markups

A document in XDOC Text consists of text with intermixed markups that describe physical and logical attributes of the text. This chapter describes the structure and syntax of the markups in an XDOC Text file.

Figure 2-1 illustrates a sample document with a variety of typefaces and paragraph indents; Figure 2-2 shows the corresponding XDOC Text file whose contents describe the sample document.

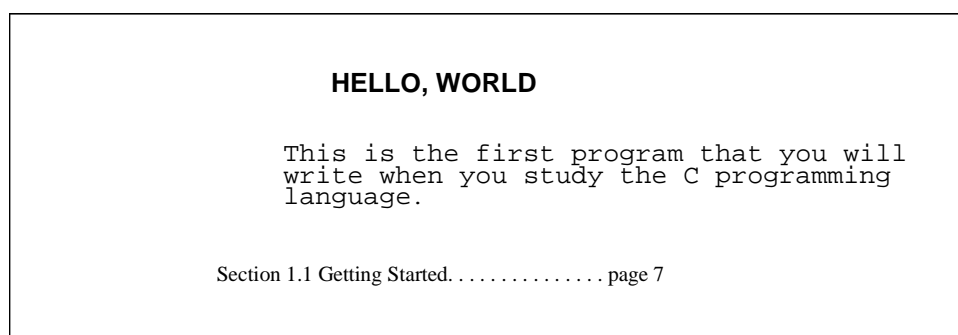


Figure 2-1. Original Page

```
[a;"XDOC.12.0";E;"FWX12.0"]
[d;"hellowconf.xdc"]
[p;1;P;0;S;0;-909;400;400;0;0;2142;2794;0;0;1]
[t;1;1;227;386;A;" ";" ";0;0;2140;2793;0;0,0,0,0]
[f;0;"<DEFAULT>";R;s;2540;F;0;0;0;10;100]
[f;2;"C";R;q;2201;F;22;21;16;9;100]
[f;3;"T";R;q;1693;V;25;25;17;10;100]
[f;4;"C";B;s;3471;F;37;37;25;15;100]

[O;1252;1]
[s;1;569;323;1;264;c;4;9][w;835]
[e;1][c;4]HELLO,
[h;1066;19;2][w;904]WORLD[y;1522;253;264;3;H]
[s;1;569;129;3;439;p;2;5][w;541][c;2]This[h;777;29;4]
[w;623]is[h;842;28;5][w;581]the[h;928;28;6][w;477]first
[h;1055;25;7][w;669]program[h;1228;25;8][w;451]that[h;133
1;25;9][w;811]you[h;1417;23;10][w;629]Will[y;1522;0;439;1
;S][s;1;569;128;11;482;p;2;5][w;601]write[h;800;24;12][w;
789]when[h;908;23;13][w;861]you[h;992;26][w;623]study[h;1
121;25;14][w;824]the[h;1204;25;15][w;492]C[h;1246;25;16][
w;733]programming[y;1522;19;482;1;S][s;1;569;130;;17;523;
p;2;5][w;526]language[y;1522;658;523;2;H][s;1;569;0;20;18
;608;t;3;0;1][w;532][c;3]Section[h;673;15;21;19][w;632]1.
1[h;723;14;22][w;601]Getting[h;841;12;23][w;632]Started[1
;. ";950;266;24;15;1][w;523]page[h;1280;12;24][w;794]7[y;
1522;215;608;0;H][g;1666;0;0;2142;2794;0]
```

Figure 2-2. XDOC Text file

2.1 Modifiers and Operands

Within XDOC, markups are called **modifiers** and the values assigned to their attributes, if any, are called **operands**. Operands appear in a list format. An operand's position in the list, rather than an inserted tag, indicates its particular meaning or attribute.

There are three basic types of operands: characters, integers, and strings. A **character operand** is a single alphabetic character. Character operands enumerate a limited number of discrete values for the operand.

Numeric operands consist of a sequence of up to 10 digits preceded by an optional minus sign. They encode quantitative data. A numeric operand cannot include a decimal point, and thus only integer values are represented.

A **string operand** is a sequence of printing characters, where the sequence can be quite large (up to 256 characters), and can include all printing characters. String operands are enclosed in double quote characters.

Note If a string operand includes the double quote character, two double quote characters are used together. In all other respects, string operands are literal. They have no embedded markups or modifiers.

All XDOC Text modifiers begin with the modifier start escape character:

[

To avoid confusion, the body of the text cannot include the modifier start escape character literally. Where [should appear in the body of the text, XDOC replaces it with the doubled modifier start escape character:

[[

The modifier code, which is a single alphabetic character, follows immediately after the modifier start escape character. Lowercase and uppercase modifier codes are available. If the modifier does not have any operands then it is followed directly by the text that it modifies otherwise it is followed by the operands. All operands are preceded by an operand separator character:

;

At the end of the operand list (but only if there is such a list) comes the modifier end escape character:

]

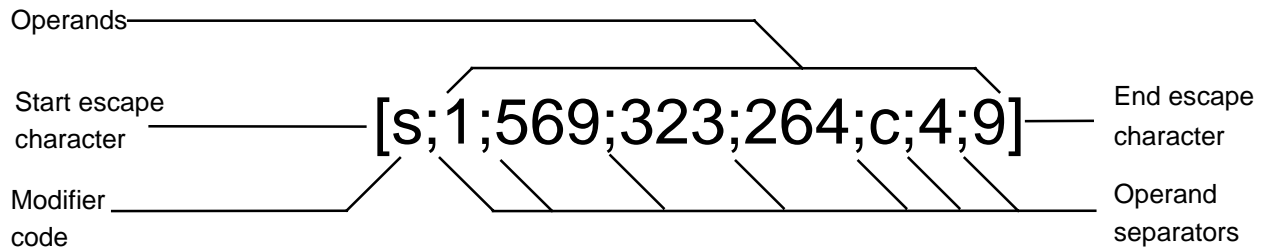


Figure 2-3. Sample XDOC Text markup

Syntactically, XDOC Text can include any text and any modifier. However, once a modifier is begun, it must be completed before a new modifier can begin.

2.2 Transactions

A **transaction** is a series of markups that, taken together, describe a block of text as a unit. There are three fundamental transactions in XDOC Text:

- the line of text
- the page
- the document

When a new transaction begins in XDOC Text, the previous transaction of the same type terminates. For example, text continues to flow into the current text line until a new text line begins. No further text is added to the previous line.

Similarly, text flows into the current page, line-by-line, until a new page begins. At that point, no new text can be added to the previous page.

Finally, text will flow into the current document, page-by-page until a new document begins. Then, no further text can be added to the previous document.

The start of the next transaction terminates the previous transaction. New data can be added to the current line, page or document until a new line, page or document has begun.

Two modifiers, one at the end of a text line and one at the end of a page, provide summary information about the previous line or page. These informational modifiers do not force the current transaction to end.

There is also an optional modifier that marks the end of the current document for your convenience. This modifier does not prevent you from adding more information to the current document.

As shown in Figure 2-4, the three transaction types are usually nested so that line transactions occur within an ongoing page transaction, and pages occur within an ongoing document transaction.

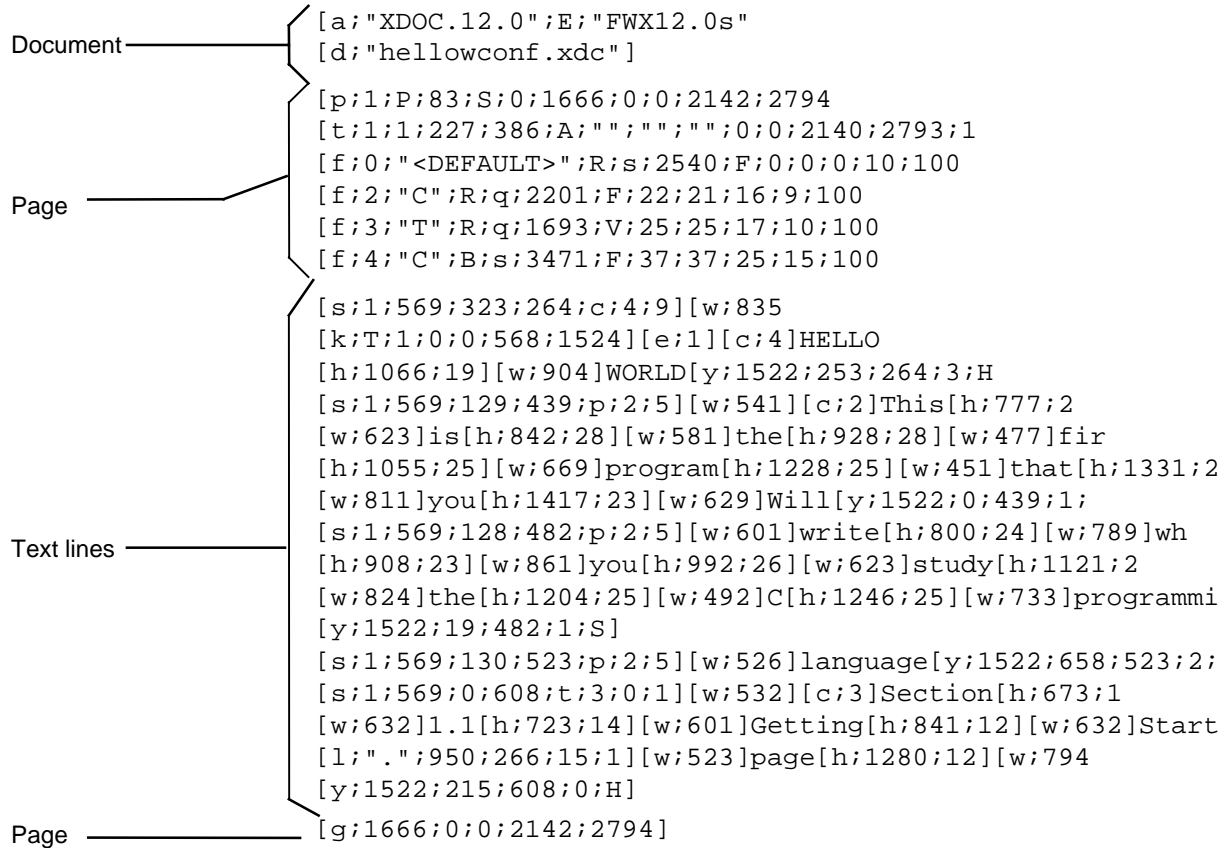


Figure 2-4. Nested XDOC transactions

Refer to Section 2.3, “Text Transaction Syntax,” for an alphabetical listing of each modifier and its associated operands.

2.2.1 Text line transactions

All text is organized into lines. Each line has a single established Y coordinate or baseline value. Words in the line of text are separated by whitespace. Each whitespace has a beginning X coordinate and a length.

There are two types of whitespace: non-printing (horizontal space) and printing (leader). Each line has a left margin and a right margin, which are, in effect, whitespaces at the beginning and end of the line between the line and the margins of the galley in which that line sits.

Modifiers for text line transactions fall into three categories: essential modifiers, mode shift modifiers, and lexical modifiers.

2.2.1.1 Essential modifiers

The essential modifiers for a text line transaction are:

Modifier	Example
start-of-text line	[s;1;569;128;5;482;p;2]
text line information	[y;1522;0;439;1;S]
horizontal space (nonprinting whitespace)	[h;842;28;3]
leader characters (printing whitespace)	[l;". ";950;266;2;15;1]page[h;1280;12;3]7

2.2.1.2 Mode shift modifiers

XDOC may apply a number of modifiers indicating a mode shift to the text within a text line. These mode shifts are:

Modifier	Example
change font	[c;1]
change region	[e;1]
change subscript	[B1[B
change superscript	[S1[S
change underline	[Umust[U
dropped capital letter	[u;763;159;234;869;1;3]

2.2.1.3 Lexical modifiers

Lexical modifiers substitute for normal text or indicate the quality of normal text. They are:

Modifier	Example
optional hyphen	[H
questionable character	[Ql
unrecognized character	[E
character confidence	[q;581]
word confidence	[w;904]
lexical class (built-in and user)	[v;14]
word bounding box	[b;892;229;1066;274;250;0]

2.2.2 Page transactions

There are three types of page transaction modifiers: essential modifiers, layout modifiers, and font modifier.

2.2.2.1 Essential modifiers

A page transaction always begins with a start-of-page modifier and ends with a page information modifier. These are the following:

Modifier	Example
start-of-page	[p;1;P;0;S;0;909;400;400;0;0;2142;2794;0;0]
page information	[g;1666;0;0;2142;2794;0]

2.2.2.2 Layout modifiers

Three modifiers provide the page layout. These are the following:

Modifier	Example
text zone descriptor	[t;1;1;227;386;A;"";"";"";0;0;2140;2793;0;1]
image zone descriptor	[x;2;11;183;1483;1916;954;"H1.tif";183;1483;1916;954]
ruling descriptor	[r;805;2584;H;1610;s;3;0;0;R;0]
new table	[j;3;251;373;2024;2268;5;5;0;261;399;399;681;681;995;995;1489;1489;2011]
start table column	[n;3;2;1;1;0;0;2;0]
end cell table	[A
hard column break	[X
section break	[k;T;2;-2;1;1;1;1;301;934;982;1610]

2.2.2.3 Font modifier

XDOC Text specifies the font information modifier for each font page-by-page even though font information is properly part of the document transaction. This means that XDOC Text may describe the same font inconsistently from page to page.

To resolve this problem, you may need to search for and use the last description of a font in a document, proceeding in workflow order. In general, you can use the font description on a page when processing that page. Any font appearing on a page has a corresponding description on that page.

There is one font information modifier.

Modifier	Example
font identifier	[f;3;"T";R;q;1693;V;25;25;17;10;100]

2.2.3 Document transactions

There are three modifiers appropriate to the document transaction. These are:

Modifier	Example
start of document	[a;"XDOC.12.0";E;"FWX12.5"]
end of document	[Z
document name	[d;"hellowconf.xdc"]

2.3 Text Transaction Syntax

Table 2-1 lists the XDOC Text modifiers alphabetically by modifier code. In each description, the modifier code is a literal code value. All operands, however, are just variable names for values, whose first letter indicates the type of the operand: c (character), n (numeric) or s (string).

The second part of the operand name is a number indicating the ordinal position of the operand. For example, "n3" is the name of the third operand; its value is a series of characters representing an integer.

The XDOC Define is the modifier name as defined in the `kdoctext.h` include file. `kdoctext.h` also defines the acceptable values for each operand.

Table 2-1. XDOC Text Markups

* indicates not found in XDOC_LITE.

Code	Meaning	Operands	XDOC Define
a	Start of document	s1: version identifier—always "XDOC12.0" c2: XDOC flavor - enhanced, lite or plus s3: OCR version - version of OCR engine	KDC_STDOC
A	End cell table	None	KDC_NDTABLE
b	Word bounding box	n1: n2: n3: n4: left, top, right, bottom bounding box of the next word n5: baseline n6: leader dot; true or false	KDC_WBOX
B	Shift to/from subscript	None	KDC_SUB
c	Change font	n1: identifier of the font changed to	KDC_CHGFONT
d	Document name	s1: internal document name, defined by user	KDC_DOCNAME

Table 2-1. XDOC Text Markups

Code	Meaning	Operands	XDOC Define
E	Unrecognized character	None	KDC_UNREC
e	Change region	n1: identifier of the region being changed to	KDC_REGION
f	Font information	n1: font identifier s2: font name c3: face style c4: serif style n5: average character width c6: code for fixed or variable character width n7: average height of uppercase characters, 0 if unknown n8: average height of lowercase characters with descender, 0 if unknown n9: average height of lowercase characters with no descender and no ascender, 0 if unknown n10: typographers point size n11: font width, e.g., 100%=normal, 80%=compressed, 120%=expanded	KDC_FONTINFO
g	Page information	n1: cosecant of angular tilt of page on image n2, n3: X,Y coordinates of top left corner of source image n4, n5: X,Y coordinates of bottom right corner of source image, c6: required or opt page break	KDC_NDPAGE
H	Optional (soft) hyphen	None	KDC_OHPHEN
h	Nonprinting whitespace	n1: X coordinate of left edge of whitespace n2: distance from left edge of whitespace to next word n3: unique id of next word *n4: space character count for tabulation from previous word (output only if more than one space) *n5: tab advance character count for tabulation from previous word (output if line is part of a table)	KDC_HSPACE

Table 2-1. XDOC Text Markups (cont.)

Code	Meaning	Operands	XDOC Define
j	New table	n1: unique id of cell table n2: n3: n4: n5: left, top, right, bottom coordinates of table n6: number of columns in table n7: number of rows in table n8: position of table on page (currently always left) n9-x: pairs of left, right coordinates for each column (x+1)-y: prs of top,bot coordinates for each row.	KDC_STABLE
J	End Drop Cap	None	KDC_NDROP
*k	Section change	c1: type (column, header, footer, caption, timestamp) n2: number of columns n3: If type is header or footer then this operand is the position (left, right, center). If type is caption then value is picture id. n4: If type is column then this operand indicates if there are vertical lines between the columns. If type is caption then value is caption expands up or down n5: number of vertical half lines to output in the default font if not caption or inset n6: use balanced columns for word processors that can handle balanced cols and hard col breaks (such as MS Word) n7: use balanced cols and hard col breaks for word processors that do not properly handle this (such as WordPerfect) n8-x: pairs of left, right coordinates for each column in the section	KDC_SECTION

Table 2-1. XDOC Text Markups (cont.)

Code	Meaning	Operands	XDOC Define
l	Printing whitespace (leader characters)	s1: repeated character in the leader n2: X coordinate of left edge of whitespace n3: distance from left edge of whitespace to next word n4: unique id of next word *n5: space character count for tabulation from previous word (output only if more than one space) *n6: tab advance character count for tabulation from previous word (output if line is part of a table)	KDC_LEADER
M	Start/stop headline	none	KDC_HEADLINE
n	Start table cell	n1: unique id of table that cell belongs to n2: current column number n3: number of columns the cell spans n4: number of rows the cell spans n5: does this cell exist or is it a continuation of a cell above or to the left n6: cell horizontal alignment (always; left) n7: number of decimal places for decimal alignment (always 2, not yet supported) n8: cell vertical alignment (always top)	KDC_SCOL
O	language	n1: MS Windows code page n2: language (see langids.h)	KDC_LANGUAGE
o	Start new table row	n1: row height (always 0) n2-x: border codes for top, left, bottom right side of each cell in the row. A border code of 0 indicates that it is a "fake" border between two sub cells (joined or spanned cells).	KDC_SROW

Table 2-1. XDOC Text Markups (cont.)

Code	Meaning	Operands	XDOC Define
p	Start-of-page	n1: logical page number c2: code for page orientation, portrait or landscape n3: recomp on or off? c4: recognition mode n5: image skew n6: page untilt—cosecant of angular correction already applied n7: x resolution n8: y resolution n9: n10: X,Y coordinates of original top left corner of page n11: page width, 0 if unknown n12: page height, 0 if unknown n13: user set or ok'd zones (M_USER_SPECIFIED_REGIONS) was set n14: user set or ok'd zone order (M_USER_SPECIFIED_ORDER) was set n15: word box units	
Q	Questionable character	None	KDC_QABLE
q	Character confidence	n1: character confidence (0 - 999)	KDC_CCONF
R	Reverse video?	n1: start or stop	KDC_REVERSE
r	Ruling descriptor	n1: n2: X,Y coordinates of the mid-point of ruling c3: code for orientation n4: total length of ruling c5: style: single, double, triple n6: thickness, 0 if unknown n7: interval, 0 if unknown n8: id n9: type (recognition or IP) n10: celltable ruling?	KDC_RULE

Table 2-1. XDOC Text Markups (cont.)

Code	Meaning	Operands	XDOC Define
S	Shift to/from superscript	None	KDC_SUPER
s	Start-of-text line	n1: zone number n2: X coordinate of the leftmost edge of the text on this line n3: distance from n2 to the beginning of the text on this line n4: unique id of next word n5: Y coordinate of baseline *c6: the current style tag: paragraph, table or center line *n7: identifier of primary font *n8: space character count for indent off the left margin (output if style is table, or if not 0) *n9: tab advance character count for tabulation off the left margin (output if style is table)	KDC_STLINE
t	Text zone descriptor	n1: zone identifier n2: output order n3: Y coordinate of the top of zone n4: height of the zone c5: plain text or table? s6: prefix s7: suffix s8: zone name, as defined by the user n9: top coordinate of the original region frame n10: left coordinate of the original region frame n11: right coordinate of the original region frame n12: bottom coordinate of the original region frame n13: top border visible? n14: left border visible? n15: bottom border visible? n16: top border visible? n17: inverse video?	KDC_TZONE

Table 2-1. XDOC Text Markups (cont.)

Code	Meaning	Operands	XDOC Define
U	Shift to/from underline	None	KDC_UNDERLINE
u	Dropped capital letter(s)	n1: n2: n3: n4: top, left, right, bottom. coordinates of the frame n5: partial word or complete word n6: number of lines drop cap has to the right of it (for future use)	KDC_DROP
v	Web link	n1-x web link	KDC_LINK
w	Word confidence	n1: word confidence (0-999)	KDC_WCONF
*X	Column break	*n1: hard or soft column break?	KDC_COL_BK
x	Image zone descriptor	n1: zone identifier n2: zone order n3: code of the graphics compression scheme used for the data. n4: Left coordinate of zone n5: Top coordinate of zone n6: Right coordinate of zone n7: Bottom coordinate of zone n8: zone name, as defined by the user *n9: Left expanded frame *n10: top expanded frame *n11: right expanded frame *n12: bottom expanded frame	KDC_PZONE
Y	Character box	n1: left n2: top n3: right n4: bottom	KDC_CBOX
y	Text line information	n1: X coordinate of the right edge of the text zone at this line n2: distance from end of the text to the right edge of zone on this line n3: Y coordinate of baseline *n4: number of line advances in current font to next line *c5: hard or soft line ending here	KDC_NDLINE

Z	End of document	None	KDC_NDDOC
---	-----------------	------	-----------

XDOC File Structure

This chapter describes the structure of an XDOC Text file. This information can help you determine how your applications should input XDOC Text data and interpret it.

Each XDOC Text file represents a separate document and, in most cases, serves as the starting point for referencing document images. An XDOC Text file does not cross-reference any other XDOC Text files.

Although further processing can combine documents or split them apart, the end result is always one or more XDOC Text files, each with its associated graphics files, and each standing independent of the others.

3.1 Page Sequence

An XDOC Text file arranges the text of a document as a sequence of pages. However, the sequence of pages in the physical XDOC Text file is not necessarily the intended presentation sequence. Each page in the document has a logical sequence number that specifies the intended presentation order, but the pages in the physical XDOC Text file occur in working order.

By concatenating individual XDOC files together, you can use a single XDOC Text file to represent multiple documents, such as the chapters of a book. The first page of each document would divide one chapter from another. If you do concatenate documents, the page numbering system must span the entire XDOC Text file.

The XDOC reader in your application might first collate the pages into ascending page number sequence, with any duplicates together. When reading an XDOC Text file with multiple documents, a page that begins a document should always be the first page of that document regardless of the collated order. All other pages in the XDOC Text file can migrate from one document to another for page collation purposes.

Each document has an internal document name. When an XDOC Text file contains only one document, as is usually the case, the internal document name is also the name of the entire document.

☞ The XDOC Text files should be stored within a Text Information Management System, where document naming issues can be resolved.

Figure 3-1 shows four documents in an XDOC Text file. The first page of the document, Chapter W, is page 1. The first page of Chapter X is page 20. The first page of chapter Y is 30. The first page of Chapter Z is page 55.

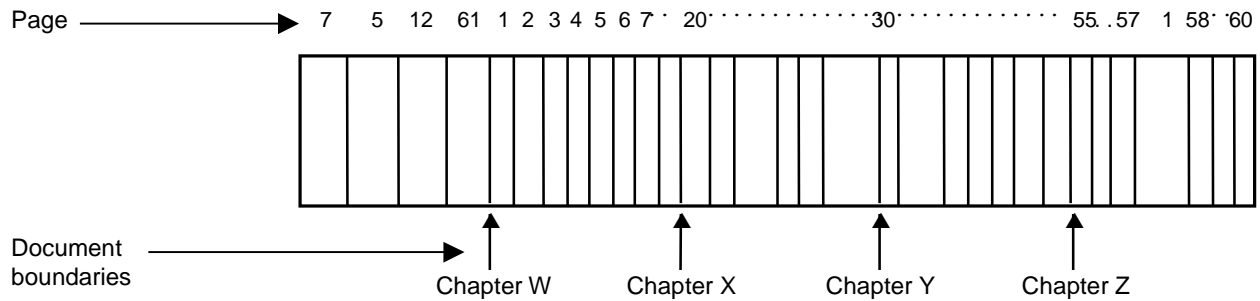


Figure 3-1. XDOC text file: macro structure

The pages of Chapter W are **not** in sequence. Pages 1, 5, and 7 are duplicated and the duplicates are out of sequence.

☞ ScanSoft’s document recognition software repeatedly attempts to recognize the data on each page until it can make a positive character identification. This means that the information in the last version of any duplicated page is always the most accurate.

3.2 Text Data

XDOC Text is character data, where each character is an eight-bit byte. The character set is a modified version of the International Standards Organization (ISO) 8859/1 character set, (see Appendix C). Extensions to the ISO set provide special characters needed for high-end publishing applications.

XDOC Text includes many newline characters that make it easier to read the character data with native-platform text display software for UNIX, DOS, or Macintosh. (Conventional conversions are available for text display on platforms other than the platform where the XDOC Text file was created.)

These newline characters are not part of the XDOC data itself. Instead, explicit XDOC Text markups indicate the organization of the source text into lines of text. XDOC Text includes no other nonprinting characters.

XDOC text uses explicit whitespace markups, rather than Space, Horizontal Tab and Vertical Tab characters to represent horizontal and vertical text spacing. Refer to Chapter 2, “XDOC Text Markups,” for more detailed information.

Store and retrieve applications may treat XDOC Text as a pure byte stream with no internal structure. Since the bytes are not limited to ASCII character codes, it may be necessary to treat these files as binary.

Applications that interpret XDOC Text must incorporate an XDOC Text parser. You can create a text parser either by direct coding or by specifying a grammar, which is then converted to a program by a standard parser generator, such as `lex` on UNIX. Because XDOC Text is a regular language, it may be parsed with a Finite State Automaton.

Page Image Analysis

XDOC Text files include detailed information about the original page image. This chapter explains how the document recognition software initially perceives the page layout, segments it into areas of graphics or text, handles rulings, and identifies fonts.

4.1 Units of Measure and Coordinate Transformations

XDOC Text data contains X,Y coordinates for text and, if known, for the page frame that contains the text. The recognition software positions the page and the text on an image with a rectangular coordinate system whose origin coincides with the upper left hand corner of the image and in which X increases to the right and Y increases downward.

The recognition software may initially perceive a page as **offset** and its text as **skewed** within the coordinate system of the image. A page that is offset is removed from the origins (0,0) of the X and Y axes of the coordinate frame. Skewed text appears to be rotated. It runs neither parallel nor perpendicular to the X and Y axes of the coordinate frame.

Figure 4-1 illustrates a page with exaggerated skew and offset.

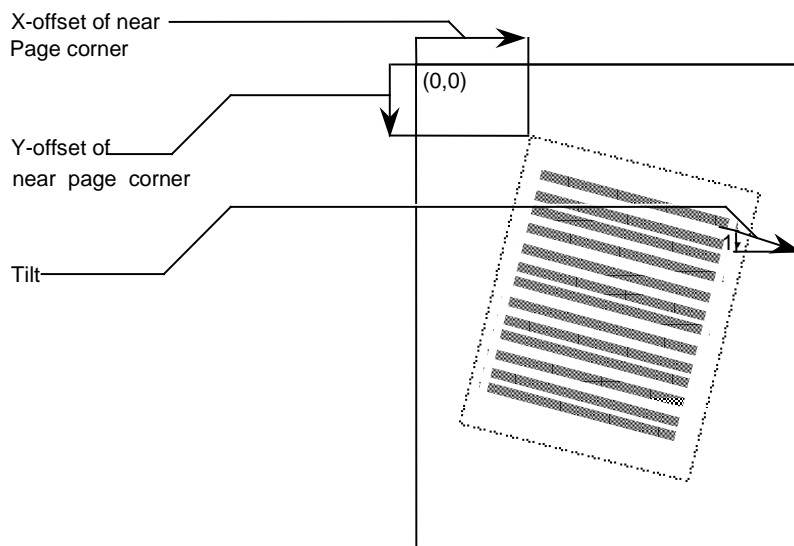


Figure 4-1. Offset and skew

The lengths of the heavy arrows represent the X offset and Y offset of the near page corner, and the **tilt**. These values are recorded in XDOC Text as part of the page description.

Tilt is the cotangent of the angle, which represents the angular skew. Tilt is the number of units of horizontal movement needed along a line of text to encounter one unit of vertical displacement.

XDOC Text never includes a tilt value of less than four. A tilt value that small would correspond to text whose angular skew is unacceptably large for the software to perform accurate recognition.

A tilt of 4000 corresponds to a negligible angular skew for pages of reasonable length and width. Consequently, very large tilts (very small angular skews) are represented by a fixed tilt value that is approximately 4000.

Since the actual text skew in acceptable document images is relatively small, the recognition software can eliminate it by approximate means without introducing noticeable errors. Once the recognition software eliminates skew, the text lines are horizontal and the left margin is vertical.

Note The recognition software determines and corrects the text skew, not the paper (page) skew.

In XDOC Text, coordinates are never actually skewed. Instead, skew is always reduced to **shear** by projecting the skewed baseline of a line of text to the left text margin for the page. See Appendix A, "Transformation Details," for more information.

Figure 4-2 shows the final translation of the image.

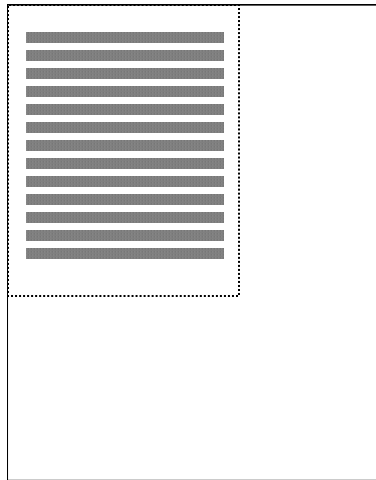


Figure 4-2. Translation of page origin to image origin
Since XDOC Text maintains values for the tilt and the translation, you can always register the text positions quite accurately with the original image.

However, since the recognition software maintains XDOC page coordinates in absolute units, and does not record the image resolution in XDOC Text, you must keep track of the scale of the text relative to the scale of the image display.

The recognition software always assumes that the image has the coordinates 0,0 for its upper left corner. You must account for any translation of the image in display. Refer to Section 4.4, "Algebraic Transformations," for more information about the transformation of page coordinates to image coordinates.

4.2 Page Segmentation

Many documents include both text and graphic images, sometimes on the same page image. These documents are called **compound documents**. The recognition software can automatically distinguish text from graphics when processing such pages and treat each type of material appropriately.

Figure 4-3 shows an illustration of a compound document.

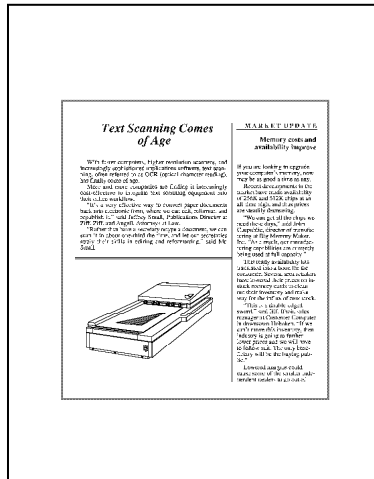


Figure 4-3. Compound document

XDOC Text describes the layout of a page in subunits called **zones**. The term zone is non-specific; it does not necessarily denote a meaningful unit of text or image. A zone can be any kind of page segment.

A **region** is one kind of page segment. A region is a defined area of the page image that is delineated by an application end user, or by the recognition software's auto-segmentation algorithms.

As shown in Figure 4-4, an application end-user can define regions within the page by drawing arbitrary, convex polygons on the page. The recognition software does not rearrange such user-defined regions by other processing.

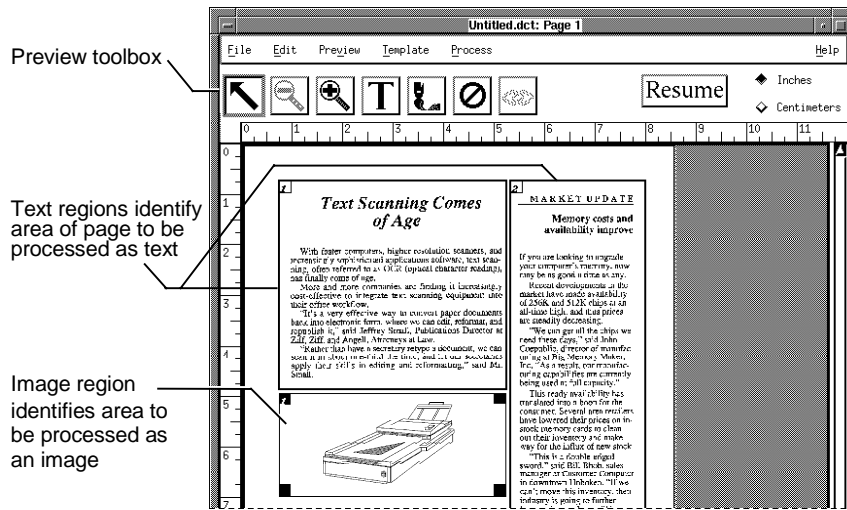


Figure 4-4. User-specified regions

A user does not need to segment compound documents manually. The recognition software automatically divides the image into regions of consistent texture.

The recognition software may then build these regions—either text or image—into XDOC Text pages, possibly one or more pages per image. The software only appropriates pages in an XDOC Text file for documents that contain running text and tabular data. Pages are holders for running text.

The recognition software does not group regions by page for tables, forms, or graphics, where physical position provides the essential context.

However, for running text, the recognition software analyzes the text regions on a page for typographic layout and document structure and links related text elements together into **galleys**.

Galley is a typographical term, meaning a unit of continuous text that includes typographic elements such as typeface, character style, and indents, but ignores physical page breaks and graphic elements.

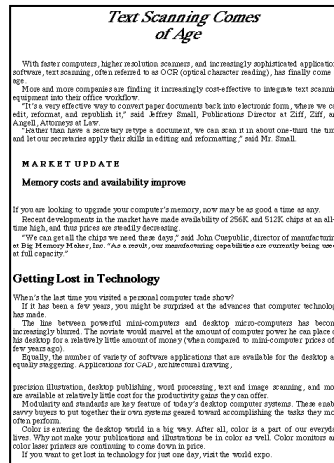


Figure 4-5. Galley

Within XDOC Text, all these varieties of page segments—user-specified regions, auto-segmented regions, linked regions, and galleys—are called zones.

☞ You can still distinguish regions, which are the fundamental unit of page segmentation within the recognition system, from other types of zones by the Change region (KDC_REGION) modifier within text zones in XDOC Text.

The recognition software never reorganizes graphics zones. A graphics zone in XDOC is identical to an image region within the recognition system.

All zones have **boundaries**. Text zones also have **margins**. Boundaries provide a framework that encloses the indicated text or graphics. The framework aids further analysis of individual zones.

The relationships between boundaries of multiple zones are relatively unconstrained. For example, zones can overlap. Also, zones need not cover the entire page. Some, if not all, of the page area can be unzoned.

For most applications, the topological layout and lexical order are more significant than the physical layout. Topological layout and lexical order are discussed in Section 4.2.1 and Section 4.2.2, respectively.

4.2.1 Topology

A text zone in XDOC Text has both a **frame** and **content**. The content is simply the list of lines of text. The XDOC Text file presents this list of text lines directly.

The recognition software derives the frame coordinates of a text zone, which are X and Y values for an enclosing rectangle, from these lines of text. The topmost and bottom-most baselines give the Y coordinates, and the rightmost and leftmost text lines (including any line margins) give the X coordinates.

The frame is not the boundary for a text zone. The left and right boundaries of a text zone are defined by the left and right extents of each line of text. This allows a text zone to have irregular sides, which run around included graphics. The top and bottom boundaries, however, are straight lines. These are identical to the top and bottom edges of the frames.

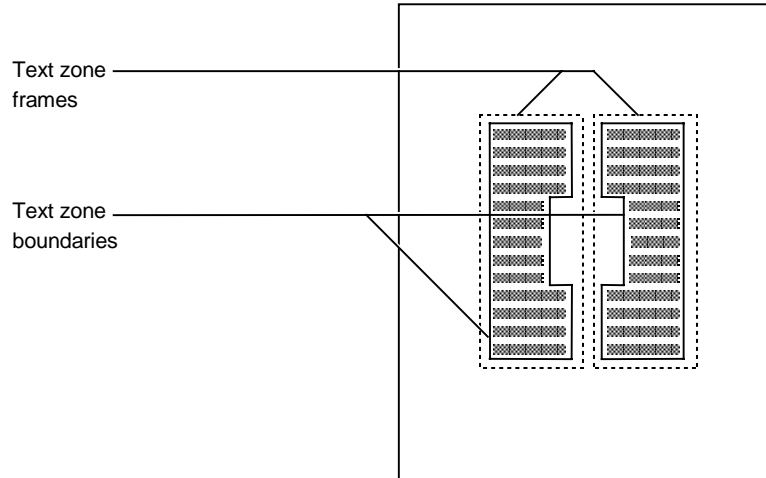


Figure 4-6. Text zone frames and boundaries

Since text zone boundaries are defined by the positions of the text lines, text zone boundaries depend on the corrections for tilt and offset that are applied to each line of text, as discussed in Appendix A.

Figure 4-7 shows the final translation of the image.

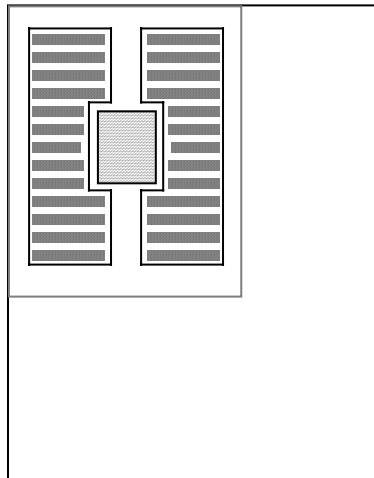


Figure 4-7. Text and picture zones on a desheared and translated page

4.2.2 Lexical order The lexical ordering of text zones preserves the reading order. Highly segmented pages can have many reading threads, which means that many lexical orderings are allowable for those text zones. No reading thread is interrupted by some other reading thread in the listed sequence of the zones.

In an XDOC Text file, the recognition software sequences the lines of text in the lexical order specified by the zone order. The lexical order of the zones is specified by the sequence of text zone frames.

4.2.3 Image zones In most respects, XDOC Text handles image zones very differently from text zones. For example, the content of each image zone is stored in a separate file. The image in this file has a rectangular frame whose sides are parallel to the axes of the original image. However, the image within that frame will be tilted if the original page image was skewed. Unlike skewed text, the recognition software does not correct image skew in any way.

The origin of an image is at the upper left corner of the frame of the image zone. XDOC treats the edges of the frame of the image zone as crop lines for displaying the image in the associated file.

4.3 Rulings

Rulings are horizontal or vertical lines on the page image. XDOC treats them like zones without text. Rulings are a property of the page but not a property of the text.

XDOC Text specifies the orientation of each ruling as horizontal or vertical. The position of a horizontal ruling is defined by the X,Y coordinate of its mid-point and its total length (X extent). Similarly, the position of a vertical ruling is given by the X,Y coordinate of its mid-point and its total length (Y extent).

The transformations for rulings from a skewed image to XDOC Text are almost exactly the same as the transformations that the recognition software uses to eliminate skew and offset in lines of text and in zone boundaries. The only difference is that the recognition software positions rulings by reference to their mid-points only.

The recognition software projects the end points of all rulings at the skew angle to the left edge of the paper (Figure 4-8) and then projects them back horizontally to their sheared position.

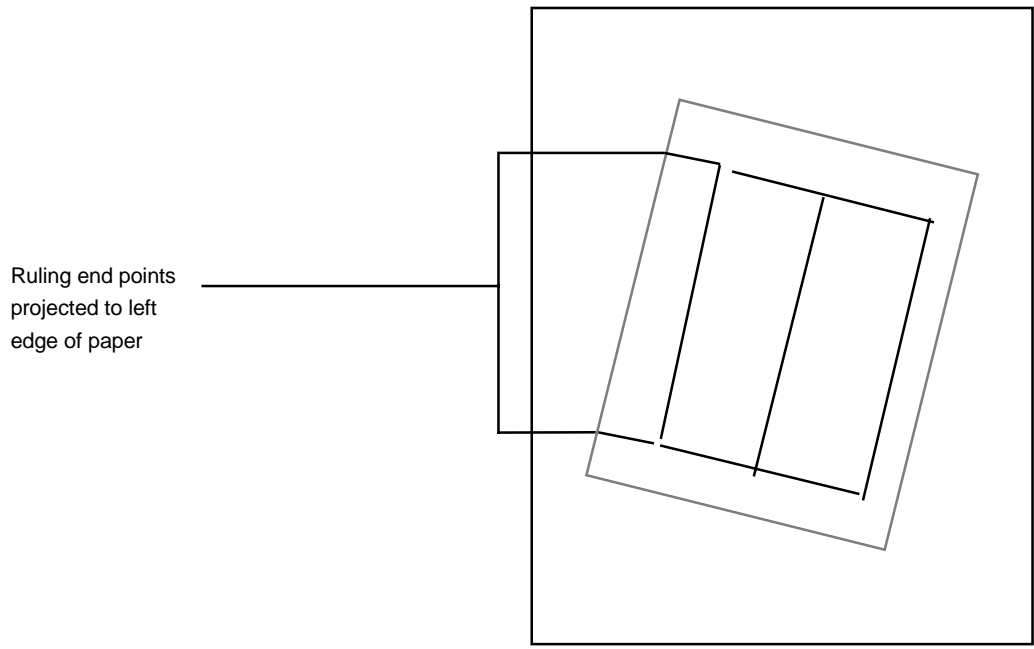


Figure 4-8. Rulings on skewed and offset page

Vertical rulings remain vertical and horizontal rulings remain horizontal. The rulings are now easily translated to the upper left of the image by translating each ruling's mid-point, as shown in Figure 4-9.

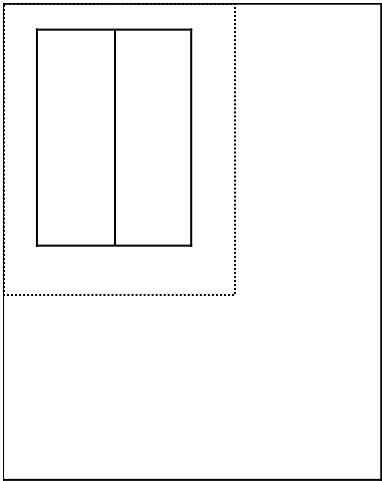


Figure 4-9. Rulings after deshearing and translation

4.4 Algebraic Transformations

This section describes specifically how to transform the following:

- Convert page coordinates to image coordinates
- Convert skew information to degrees
- Convert original image coordinates to deskewed coordinates

4.4.1 Convert page coordinates to image coordinates

The approximations used to transform image coordinates to the page coordinates in XDOC Text introduce negligible errors. To perform the reverse transformation, from page coordinates to image coordinates, you can use the classical trigonometric affine transformation to scale, rotate, and translate the page coordinates.

Furthermore, since the skew angle is almost always quite small, you can use small angle approximations in this transformation. This simplifies trigonometric functions to arithmetic functions.

XDOC Text indirectly provides the X,Y coordinates of many interesting points on a page, for example, the X,Y coordinates of the lower left corner of an arbitrary word.

If that word is the first on its text line, the X coordinate is the sum of the second and third operands of the Start-of-line (KDC_STLINE) modifier. For any other word in that line, the X coordinate is the sum of the first and second operands of the preceding Nonprinting whitespace (KDC_HSPACE) modifier or Leader character (KDC_LEADER) modifier.

The Y coordinate of this point is the sum of the baseline for this text line and the length of the average descender for the primary font of the line. The baseline appears in both the fourth operand of the KDC_STLINE modifier and the third operand of the Text line information (KDC_NDLINE) modifier.

The length of a descender for the primary font of the line is the eighth operand less the ninth operand of the Font information (KDC_FONTINFO) modifier for that primary font. The primary font for the text line is given in the sixth operand of the KDC_STLINE modifier for that line.

The coordinates of the other corners of the word (in the page coordinate system) are obtained as follows:

- The Y coordinate of the lower right corner is identical to the Y coordinate of the lower left corner, which was described above.
- The Y coordinates of the two upper corners are also identical. Their value is the baseline value minus the height of large characters in the primary font of the line. This height is the seventh operand of the KDC_FONTINFO modifier for the primary font of the line.
- The X coordinate of the upper left corner is identical to the X coordinate of the lower left corner.

- The X coordinates of the two left corners are also identical. If the word is the last word in the line, this X coordinate value is the first operand less the second operand in the KDC_NDLINe modifier of this line.
- For all other words in this text line, the value of the X coordinate of the left corners is the first operand of the immediately following KDC_HSPACE modifier, or the second operand of the following KDC_LEADER modifier.

Let $(\Delta x, \Delta y)$ be the coordinates of the upper left-hand corner of the frame for the current page. The upper left corner and the dimensions of this page, if known, are given in the fifth through eighth operands of the Start page (KDC_STPAGE) modifier. (Otherwise, the image frame, as given in the Page information (KDC_NDPAGE) modifier, is taken as the page frame.)

The unit of measure of all coordinates in XDOC Text is 0.1 millimeters (mm). All coordinates in the page of text must be scaled to the size of the display image. Multiply each X coordinate and Y coordinate by a constant, which is the number of units of distance on the displayed image needed to represent 0.1 mm on the source image.

Let (X, Y) be the arbitrary point (x, y) after scaling and let $(\Delta X, \Delta Y)$ be the position of the upper left hand corner of the page on the image after scaling. The cotangent of the angular skew of the text within the image is the first operand of the KDC_NDPAGE modifier. Let T (for tilt) be this cotangent.

Since the angular skew is small, the tangent of the angular skew is small and approximately equal to the sine of the angular skew. The cosine of the angular skew is very close to 1.0. Thus, the following simple, approximate (but rather accurate) coordinate transformation carries (X, Y) to $(\underline{X}, \underline{Y})$, the equivalent image coordinate.

$$\underline{X} = X - Y/T + \Delta X$$

$$\underline{Y} = Y + \Delta Y$$

The design intent of XDOC is to describe, to a word processor, a page as seen by recognition. The X values found in XDOC output very accurately position “word boundaries” (begin word, end word).

However, the Y values are chosen from among the character Y values, such that a coherent horizontal line of text will be rendered. A Y value, somewhere between the maximum and minimum, is derived using proprietary heuristics. Thus, the horizontal placement of a given Y value cannot be assumed to be at any specific position on the horizontal line.

Note When transforming page coordinates to image coordinates on a multi-zone document, you should process each zone separately rather than applying transformations to the page as whole. Refer to Appendix A, “Transformation Details,” for more in-depth information.

4.4.2 Convert skew information to degrees

The angular correction of the text on a page can be retrieved from XDOC from the KDC_STPAGE markup or the KDC_NDPAGE markup.

The following is an example of a KDC_STPAGE markup with its operands:

```
[p;1;P;0;S;0;-20;400;400;0;0;2142;2794;0;0;1]
```

The fourth operand, (-20) represents the cosecant of the angular correction already applied to the text. This value can be positive or negative depending on the angle of correction. To convert this value to degrees do the following:

$$\arcsin(1/\text{cosecant}) = \text{angle of correction}$$

Table 4-1 shows some examples of this transformation.

Table 4-1. Sample Transformations

Cosecant	Calculated Angle of Correction	Perceived Tilt
4000	0.014 degrees	insignificant
200	0.28 degrees	barely tilted
-30	-1.9 degrees	quite tilted
20	2.8 degrees	quite tilted
12	4.7 degrees	severe tilt

4.4.3 Convert original image coordinates to deskewed coordinates

The V_XDC_WBOX tagged value controls the output of word bounding box coordinates in XDOC format. When the value of this tag is non-zero, the API outputs word-bounding boxes for all word whose confidence is less than or equal to the value of V_ACCEPT_THRESH. The default value is 0.

☞ To include bounding boxes for all words, set the value of V_ACCEPT_THRESH to 999.

Word bounding boxes enable you to locate and highlight words on the page image seen by recognition. Word bounding boxes are passed to the text output system from recognition; boundaries include ascenders and descenders, where applicable. Because the ascenders and descenders are included, you cannot use bounding box coordinates to determine the word baseline.

If you want to use word bounding box coordinates to refer to the image **after** recognition, you must save the image if any preprocessing, such as rotation or deskewing, was performed. The coordinates match the image passed to recognition from preprocessing.

However, if the deskewed image is not available, this section documents the formula used to map coordinates back and forth from the original bitmap to the deskewed bitmap when an image has been deskewed using the preprocessing routines (PP_SKEW, PP_SKEW_C).

Let w and h be the width and height of the bitmap respectively.

Assume a coordinate system where the pixel at the center of the bitmap, at $(w/2, h/2)$, is the origin and all coordinates are referenced as offsets from this origin. Increasing y is up, increasing x is to the right.

The angle θ is measured in the clockwise positive sense.

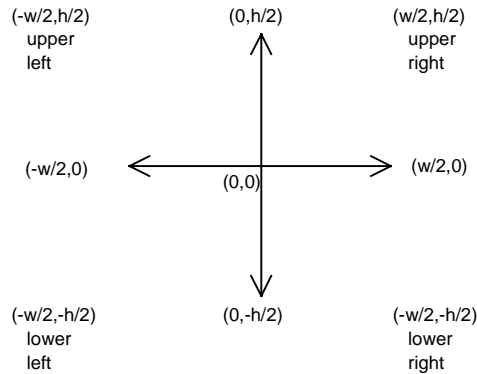


Figure 4-10. Deskewing coordinate system

To find the input pixel that got moved to a given output coordinate in terms of the above coordinate system :

$$\begin{aligned} x_{in} &= x_{out} - y_{out} * \sin(\theta) \\ y_{in} &= x_{out} * \sin(\theta) + y_{out} \end{aligned}$$

To find the output coordinate that a given input pixel will be relocated to :

$$\begin{aligned} x_{out} &= (x_{in} + y_{in} * \sin(\theta)) / (1 + \sin^2(\theta)) \\ y_{out} &= (-x_{in} * \sin(\theta) + y_{in}) / (1 + \sin^2(\theta)) \end{aligned}$$

Note, that this is different from a true rotation in which :

$$\begin{aligned} x_{in} &= x_{out} * \cos(\theta) - y_{out} * \sin(\theta) \\ y_{in} &= x_{out} * \sin(\theta) + y_{out} * \cos(\theta) \end{aligned}$$

4.5 Fonts

A **font** is a set of typeface characters of a particular design and size.

The recognition software uses letter sizes and letter shapes distinguish one font from another font. After encountering a reasonable number of letters of a consistent size and shape, the software assigns a numeric font identifier to this letter category. This font identifier is greater than or equal to 1.

If the recognition software sees too few letters in a particular size category, it assigns those characters to the numeric identifier 0. The 0 font identifier has no corresponding font description and indicates that these characters do not belong to any particular font.

The recognition software has no mechanism for merging categories once they have been identified. Thus, you cannot be certain that two similarly sized fonts are actually identical, although you can make that assumption in the interests of simplicity.

A font description consists of four basic types of data:

- font family name,
- font style (serif, italic, roman, bold),
- font spacing (mono-spaced vs. variable-spaced)
- font size

The recognition software measures font size in three different ways. The first is column width. Column width is valid only for mono-spaced fonts.

The second is in the approximate typographers point size for the font face, in units that are 72 points to the inch. Point size is rounded off to the nearest popular point size: 6, 7, 8, 10, 12, 14, 18, 24, and 36.

The third size measurement is a set of four values for the heights of four separate types of characters: uppercase, lowercase with no ascender or descender, lowercase with an ascender, and lowercase with a descender.

Figure 4-11 (next page) illustrates the distances used to obtain the four character height values for font values.

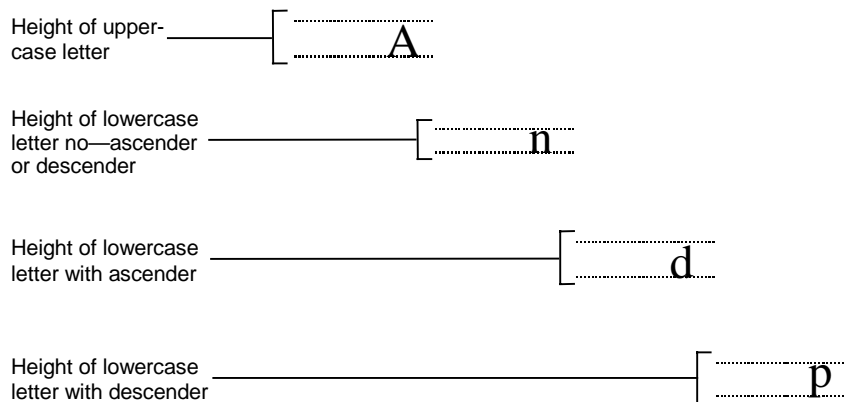


Figure 4-11. Letter heights

The recognition software gathers these values by averaging the measurement of individual characters in the designated category for the particular font. In most cases, the number of samples is quite large. The unit of font size measurement, outside of the typographers point size, is 0.1 millimeters (mm).

Transformations

This appendix provides additional details about the transformation of image coordinates to page coordinates in XDOC Text.

A.1 Text Lines

In XDOC Text, coordinates are never actually skewed. Instead, skew is always reduced to shear by projecting the skewed baseline of a line of text to the left text margin for the page. See Figure A-1 and Figure A-2.

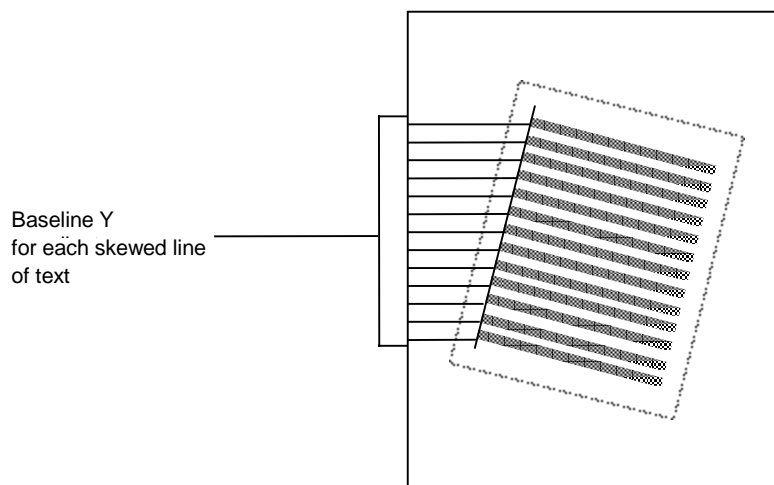


Figure A-1. Baseline Y coordinates of skewed text

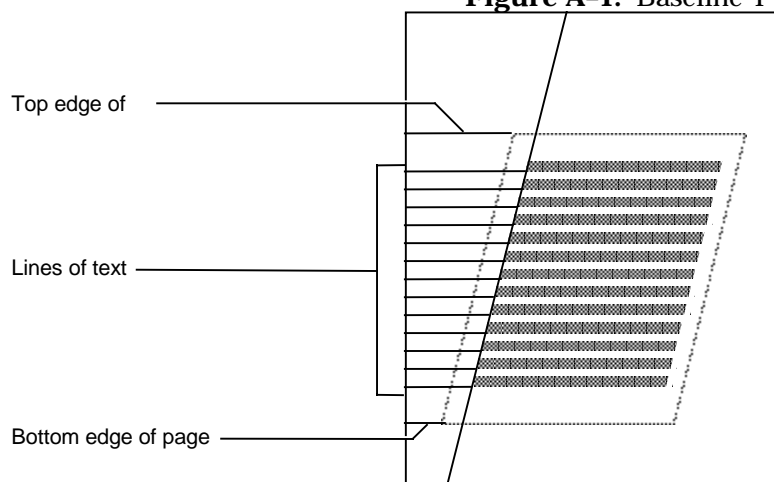


Figure A-2. Sheared approximation to skew

Thus, each line of text has a single Y coordinate, or baseline value. Lines of text in XDOC always appear as horizontal.

The document recognition software deshears all X coordinates in XDOC Text. This means that vertical text margins, such as the left margin, appear vertical. This process is illustrated in Figure A-3.

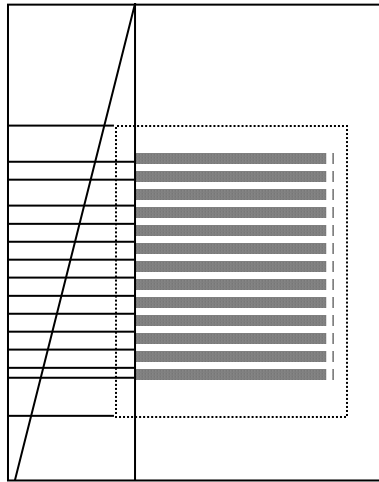


Figure A-3. Desheared lines of text

A.2 Zones

Figures A-4, A-5, and A-6 illustrate the process of transforming an offset and skewed page image to a desheared page.

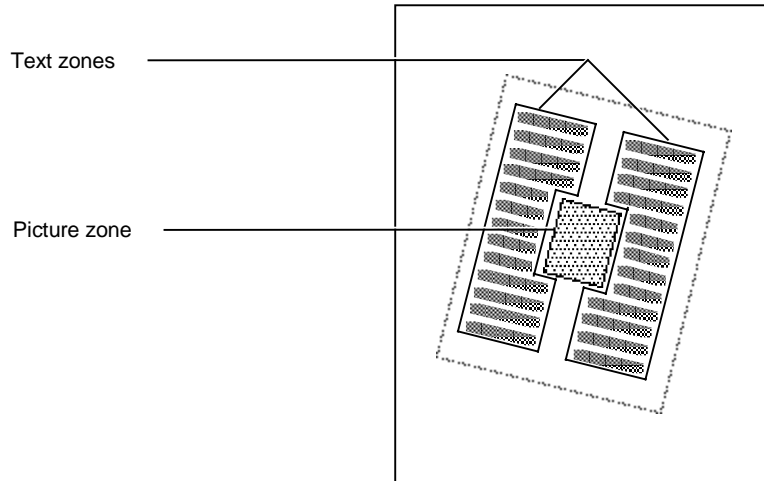


Figure A-4. Text zones and a picture zone on a skewed and offset page

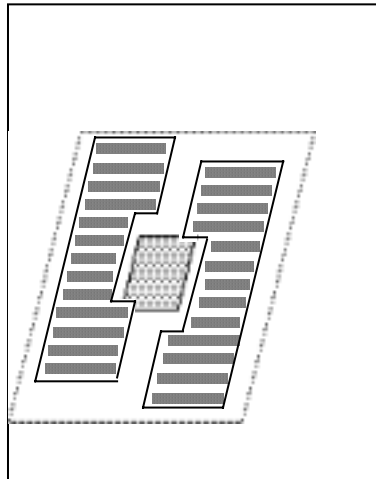


Figure A-5 Skew converted to shear, zone by zone

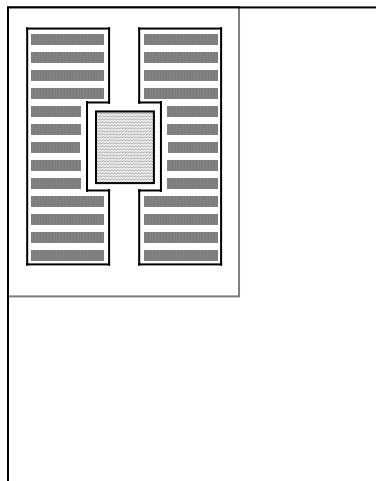


Figure A-6. Text and picture zones on a desheared and translated page

XDOC Text portrays the lines of text as horizontal and gives them the Y coordinate where the tilted line of text intersects the left text margin for the text zone.

A.3 Rulings

The first step in the transformation of a skewed image to XDOC Text is the conversion of skew to shear. This happens before the recognition software produces XDOC Text. In the case of rulings, which are page-wide features, the software performs shearing relative to the left edge of the page.

Vertical rulings are always vertical, even on a sheared XDOC Text page. This creates gaps at the junctions of horizontal and vertical rulings. However, once the software has corrected shear, these gaps disappear.

The software projects the end points of all rulings at the skew angle to the left edge of the paper and then projects them back horizontally to their sheared position (Figure A-7).

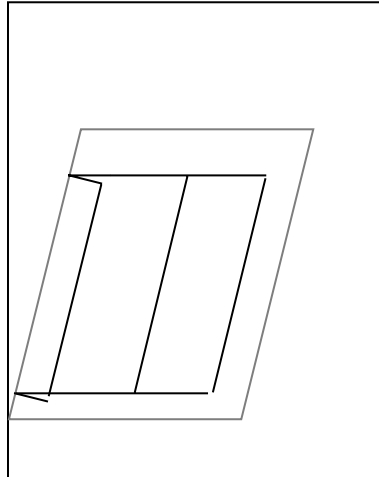


Figure A-7. Rulings converted from skew to shear at end points

From these positions, the software calculates the positions of the mid-points (Figure A-8). The vertical rulings never appear to be sheared since the software only presents the mid-points. The sheared rulings still have gaps.

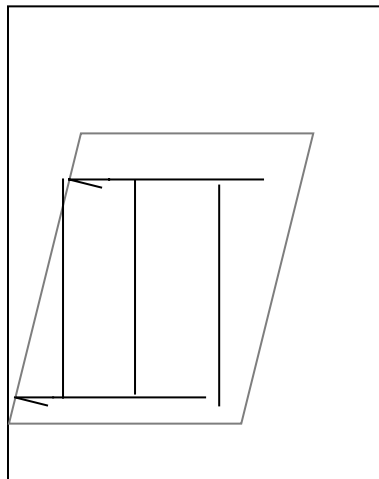


Figure A-8. Sheared rulings when presented at mid-points only

After deshearing, however, as shown in Figure A-9, the software restores the horizontal and vertical rulings to their correct relative positions. The software performs the deshearing, which is a horizontal displacement proportional to the Y coordinate of a point, at the mid-point of each ruling.

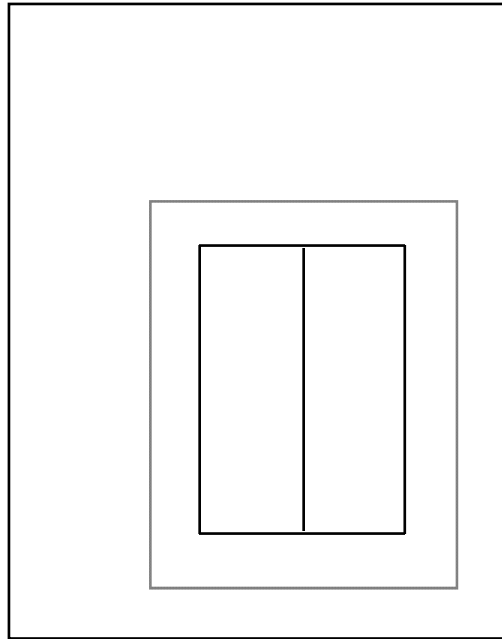


Figure A-9. Rulings after deshearing

Sample XDOC Text

This appendix provides a sample page image and its corresponding XDOC Text file.

B.1 Sample Page Image

This document has multiple columns containing alphanumeric data, telephone numbers, and currency amounts. It includes several typefaces and styles.

<i>New England Begonia Society Annual Fund Donations</i>			
Name	Address	Telephone	Donation
Peter Adams	1234 Rock Lane, Bath, ME 01201	207-555-2314	25.00
John Albert	321 Riley Road, Bath, ME 01201	207-555-3425	15.00
Patricia Arnel	822 Martin Drive, Bath, ME 01201	207-555-7869	30.0
Heidi Barnett	666 Billiard Ave., Bath, ME 01201	207-555-4454	65.00
Rocky Balboa	565 Bliss Way, Bath, ME 01201	207-555-1357	20.00
Scott Booker	787 Hunan Avenue, Bath, ME 01201	207-555-2465	25.00
Henry Brown	111 Roberts Drive, Bath, ME 01201	207-555-2233	100.00
Andrew Chate	567 Rodney Lane, Bath, ME 01201	207-555-5689	10.00
Andrea Challen	88 Patterson Road, Bath, ME 01201	207-555-0987	25.00
William Davis	5 Broadway, Bath, ME 01201	207-555-1212	35.00
Adrian Davis	32 Whitney Place, Bath, ME 01201	207-555-6935	50.00
Howard Dundee	9546 Light Avenue, Bath, ME 01201	207-555-0283	15.00
Ralph Ester	35 Togo Way, Bath, ME 01201	207-555-3729	25.00
Robert Ezra	456 Outland Avenue, Bath, ME 01201	207-555-0447	35.00
Jane Farquar	44 Peters Road, Bath, ME 01201	207-555-9742	20.00
Bill Fuzell	5 Rockpoint Road, Bath, ME 01201	207-555-8080	15.00
Jake Fratz	695 Neary Road, Bath, ME 01201	207-555-0865	25.00
Joe Gunda	900 Finn Avenue, Bath, ME 01201	207-555-2499	25.00
B. G. Hunter	12 Ryder Lane, Bath, ME 01201	207-555-1009	150.00
Peter Jacobs	33 PeeWee Way, Bath, ME 01201	207-555-9989	30.00
Paul Jacoby	7899 Helter Drive, Bath, ME 01201	207-555-3343	35.00
Dorothy Johnson	46 Main Street, Peabody, MA, 01970	207-555-4318	25.00
Kenneth Kuntz	2 Cynidy Avenue, Bath, ME 01201	207-555-9823	10.00
Jacqueline Kitee	567 Cassia Drive, Bath, ME 01201	207-555-0978	10.00
Chen Ka	545 Built Road, Bath, ME 01201	207-555-3719	30.00
Harold Langlois	200 Wayland Avenue, Bath, ME 01201	207-555-1947	25.00
William Lyttle	43 Sudbwy Drive, Bath, ME 01201	207-555-1009	35.00
Rawley Hodler	799 Concord Lane, Bath, ME 01201	207-555-2357	25.00
George Smyth	1234 Wash Ave, Peabody, MA 01970	207-555-1212	10.00

Figure B-1. Sample page image

B.2 Sample XDOC Text

This XDOC Text file includes all required markups only. The lines have been broken artificially for readability.

```
[a;"XDOC.12.0";E;"FWX12.5"]
[d;"beth.xdc"]
[p;1;P;1;S;0;-434;400;400;0;0;2150;2794;0;0;1]
[t;2;1;227;386;A;"";"";"";0;0;2140;2793;0;0;0;0;1]
[r;1069;2;H;2140;s;1;0;0;R;0]
[f;0;"<DEFAULT>";R;s;2540;F;0;0;0;10;100]
[f;2;"T";T;q;2963;V;34;34;24;14;100]
[f;3;"T";B;q;2624;V;30;29;21;12;100]
[f;4;"C";R;q;1947;F;22;21;16;9;100]
[O;1252;1]
[s;2;244;522;1;275;c;2;17]
[k;T;1;-2;1;1;1;1;243;1887][e;2][c;2]New[h;854;14;2]England[h;1045;12;3]
Begonia[h;1224;16;4]Society[y;1883;501;275;1;H]
[s;2;244;580;5;327;c;2;19]Annual[h;980;13;6]Fund[h;1105;10;7]Donations
[y;1883;558;327;2;H]
[s;2;244;129;7;469;t;3;4;1][c;3]Name[h;474;452;8;17;1]Address
[h;1070;377;9;14;1]Telephone[h;1633;84;10;3;1]Donation[y;1883;0;469;2;H]
[s;2;244;5;11;543;t;4;0;1][c;4]Peter[h;351;24;12]Adams[h;477;158;8;1]1234
[h;712;27;13]Rock[h;822;25;14]Lane,[h;941;32;15]Bath,[h;1068;31;16]ME[h;1139;
27;17]01201
[h;1265;133;18;6;1]207-555-2314[h;1648;103;19;5;1]25.00[y;1883;32;543;2;H]
[s;2;244;3;20;592;t;4;0;1]John[h;331;22;21]Albert[h;478;156;22;8;1]321[h;691;
26;23]Riley
[h;822;23;24]Road,[h;940;33;25]Bath,[h;1068;31;26]ME[h;1139;27;27]01201
[h;1265;133;28;6;1]207-555-3425[h;1648;104;29;5;1]15.00[y;1883;32;592;2;H]
[s;2;244;5;30;649;t;4;0;1]Patricia[h;414;24;31]Arnel[h;541;93;32;4;1]822
[h;689;27;33]Martin[h;843;24;34]Drive,[h;982;33;35]Bath,[h;1110;31;36]ME[h;11
81;27;37]01201
[h;1307;91;38;4;1]207-555-7869[h;1647;104;39;5;1]30.0[y;1883;54;649;2;H]
[s;2;244;3;40;704;t;4;0;1]Heidi[h;348;27;41]Barnett[h;521;113;42;5;1]666
[h;690;27;43]Billiard[h;885;22;44]Ave.,[h;1004;33;45]Bath,[h;1131;32;46]ME
[h;1203;27;47]01201[h;1328;70;48;3;1]207-555-4454[h;1648;104;49;5;1]65.00
[y;1883;32;704;2;H]
[s;2;244;3;50;760;t;4;0;1]Rocky[h;353;24;51]Balboa[h;500;135;52;6;1]565[h;691
;27;53]Bliss
[h;819;25;54]Way,[h;920;32;55]Bath,[h;1047;31;56]ME[h;1118;27;57]01201
[h;1244;155;58;7;1]207-555-1357[h;1648;103;59;5;1]20.00[y;1883;31;760;2;H]
[s;2;244;4;60;817;t;4;0;1]Scott[h;349;27;61]Booker[h;500;134;62;6;1]787[h;690
;27;63]Hunan
[h;822;23;64]Avenue,[h;983;32;65]Bath,[h;1110;31;66]ME[h;1182;27;67]01201
[h;1308;91;68;4;1]207-555-2465[h;1648;103;69;5;1]25.00[y;1883;32;817;2;H]
[s;2;244;3;70;873;t;4;0;1]Henry[h;352;24;71]Bzada[h;479;155;72;7;1]111
[h;691;26;73]Roberts[h;862;26;74]Drive,[h;1004;33;75]Bath,[h;1132;31;76]ME
[h;1203;27;77]01201[h;1329;70;78;3;1]207-555-2244
[h;1648;82;79;4;1]100.00[y;1883;32;873;2;H]
[s;2;244;2;80;929;t;4;0;1]Andrew[h;373;24;81]Chate[h;500;134;82;6;1]567
[h;690;27;83]Rodway[h;844;24;84]Lane,[h;961;33;85]Bath,[h;1088;32;86]ME[h;116
0;27;87]01201
[h;1286;113;88;5;1]207-555-5689[h;1648;103;89;5;1]10.00[y;1883;32;929;2;H]
[s;2;244;2;90;986;t;4;0;1]Andrea[h;371;25;91]Challen[h;544;90;92;4;1]88
[h;669;29;93]Patterson[h;884;25;94]Road,[h;1004;33;95]Bath,[h;1131;32;96]ME
[h;1203;27;97]01201[h;1328;71;;98;3;1]207-555-0987[h;1648;102;99;5;1]25.00
```


[y;1883;32;986;2;H]
[s;2;244;2;100;1041;t;4;0;1]William[h;396;23;101]Davis[h;520;114;102;5;1]5
[h;648;28;103]Broadway,[h;856;33;104]Bath,[h;984;31;105]ME[h;1055;27;106]0120
1
[h;1180;220;107;11;1]207-555-1212[h;1648;104;108;5;1]35[h;1787;1;1093]-
[h;1804;13;110]00
[y;1883;31;1041;2;H]
[s;2;244;2;111;1097;t;4;0;1]Adrian[h;373;24;112]Davis[h;499;135;113;6;1]32
[h;669;27;114]Whitney[h;844;25;115]Place,[h;983;33;116]Bath,[h;1111;31;117]ME
[h;1182;27;118]01201
[h;1307;92;119;4;1]207-555-
6935[h;1649;103;120;5;1]50.00[y;1883;31;121;1097;2;H]
[s;2;244;3;122;1153;t;4;0;1]Howard[h;373;24;123]Dundee[h;521;113;124;5;1]9546
[h;712;28;125]LightAvenue[h;968;69;126]Bath,[h;1131;32;127]ME[h;1203;27;128]0
1201
[h;1328;71;129;3;1]207-555-
0283[h;1647;105;130;5;1]15.00[y;1883;31;131;1153;2;H]
[s;2;244;2;132;1209;t;4;0;1]Ralph[h;351;25;133]Ester[h;478;156;134;8;1]35[h;6
69;27135]Togo
[h;777;25;136]Way,[h;876;33;137]Bath,[h;1004;31;138]ME[h;1075;27;139]01201
[h;1201;198;140;10;1]207-555-
3729[h;1648;102;141;5;1]25.00[y;1883;31;1209;2;H]
[s;2;244;2;142;1266;t;4;0;1]Robert[h;371;26;143]Ezra[h;478;155;144;7;1]456
[h;690;27;145]Outland[h;863;23;146]Avenue,[h;1025;32;147]Bath,[h;1152;31;148]
ME
[h;1223;27;149]01201[h;1350;49;150;2;1]207-555-0447[h;1647;104;151;5;1]35.00
[y;1883;32;1266;2;H]
[s;2;244;3;152;1323;t;4;0;1]Jane[h;328;26;153]Farguar[h;500;135;154;6;1]44[h;
670;29;155]Peters[h;819;26;156]Road,[h;941;33;157]Bath,[h;1068;32;158]ME[h;11
40;27;159]01201
[h;1265;134;160;6;1]207-555-9742[h;1647;103;161;5;1]20.00[y;1883;32;1323;2;H]
[s;2;244;4;162;1379;t;4;0;1]Bill[h;327;28;163]Fuzell[h;477;158;164;8;1]5
[h;648;27;165]Rockpoint[h;862;26;166]Road,[h;983;33;167]Bath,[h;1110;32;168]M
E[h;1181;28;169]01201[h;1307;92;170;4;1]207-555-8080[h;1648;103;171;5;1]15.00
[y;1883;32;1379;2;H]
[s;2;244;3;172;1436;t;4;0;1]Jake[h;328;26;173]Fratz[h;455;180;174;9;1]695[h;6
91;26;175]Neary
[h;822;23;176]Road,[h;940;33;177]Bath,[h;1068;31;178]ME[h;1139;27;179]01201
[h;1264;135;180;6;1]207-555-0865[h;1648;102;181;5;1]25.00[y;1883;32;1436;2;H]
[s;2;244;2;182;1493;t;4;0;1]Joe[h;306;25;183]Gunda[h;435;199;184;10;1]900[h;6
90;28;185]Finn
[h;799;23;186]Avenue,[h;961;33;187]Bath,[h;1089;31;188]ME[h;1160;27;189]01201
[h;1285;113;190;5;1]207-555-2499[h;1647;103;191;5;1]25[h;1786;13;192]-
[h;1803;13;193]00[y;1883;32;1493;2;H]
[s;2;244;2;194;1549;t;4;0;1]B.G.[h;322;31;195]Hunter[h;477;157;196;8;1]12[h;6
68;28;197]Ryder
[h;798;26;198]Lane,[h;918;33;199]Bath,[h;1046;31;200]ME[h;1117;27;201]01201
[h;1243;155;202;7;1]207-555-1009[h;1647;82;203;4;1]150.00[y;1883;32;1549;2;H]
[s;2;244;4;204;1604;t;4;0;1]Peter[h;350;25;205]Jacobs[h;498;136;206;6;1]33[h;
668;30;207]PeeWee[h;820;25;208]Way,[h;919;33;209]Bath,[h;1047;31;210]ME[h;111
8;27;211]01201
[h;1243;156;212;8;1]207-555-9989[h;1648;103;213;5;1]30.00[y;1883;32;1604;2;H]
[s;2;244;4;214;1661;t;4;0;1]Paul[h;326;27;215]Jacoby[h;480;153;216;7;1]7899
[h;711;27;217]Helter[h;862;25;218]Drive,[h;1003;33;219]Bath,[h;1131;31;220]ME
[h;1202;27;221]01201[h;1328;71;222;3;1]207-555-3343[h;1647;104;223;5;1]35.00
[y;1883;32;1661;2;H]
[s;2;244;1;224;1717;t;4;0;1]Dorothy[h;394;24;225]Johnson[h;564;69;2263;1]46

[h;669;26;227]Main[h;778;25;228]Street,[h;940;34;229]Peabody,[h;1131;31;230]M
A,
[h;1216;35;231]01970[h;1349;49;232;2;1]207-555-4318[h;1648;103;233;5;1]25.00
[y;1883;32;1717;2;H]
[s;2;244;1;234;1772;t;4;0;1]Kenneth[h;393;25;235]Kuntz[h;519;113;236;5;1]2[h;
646;28;237]Cyndy
[h;778;22;238]Avenue,[h;940;32;239]Bath,[h;1067;31;240]ME[h;1138;27;241]01201
[h;1264;134;242;6;1]207-555-9823[h;1646;105;243;5;1]10.00[y;1883;32;1772;2;H]
[s;2;244;0;244;1830;t;4;0;1]Jacqueline[h;456;26;245]Kitee[h;584;49;246;2;1]56
7
[h;689;27;247]Cassia[h;841;25;248]Drive,[h;981;33;249]Bath,[h;1109;31;250]ME[
h;1180;27;251]01201
[h;1307;91;252;4;1]207-555-0978[h;1647;104;253;5;1]10.00[y;1883;32;1830;2;H]
[s;2;244;1;254;1886;t;4;0;1]Chen[h;329;25;255]Ka[h;392;242;256;12;1]545[h;690
;27;257]Built
[h;819;26;258]Road,[h;940;33;259]Bath,[h;1067;32;260]ME[h;1139;27;270]01201
[h;1264;135;271;6;1]207-555-3719[h;1648;104;272;5;1]30.00[y;1883;32;1886;2;H]
[s;2;244;1;273;1941;t;4;0;1]Harold[h;372;25;274]Langlois[h;562;70;275;3;1]200
[h;690;25]Wayland[h;864;22]Avenue,[h;1025;33]Bath,[h;1152;32]ME
[h;1224;27;276]01201[h;1349;49;277;2;1]207-555-1947[h;1647;104;278;5;1]25.00
[y;1883;32;1941;2;H]
[s;2;244;0;279;1999;t;4;0;1]William[h;394;24;280]Lyttle[h;541;91;281;4;1]43
[h;667;30;282]Sudbury[h;843;23;283]Drive,[h;982;33;284]Bath,[h;1109;31;285]ME
[h;1180;27;286]01201
[h;1307;91;287;4;1]207-555-1009[h;1647;104;288;5;1]35.00[y;1883;32;1999;2;H]
[s;2;244;1;289;2055;t;4;0;1]Rawley[h;372;23;290]Mooler[h;520;112;291;5;1]799
[h;689;28;292]Concord[h;863;25;293]Lane,[h;982;33;294]Bath,[h;1109;32;295]ME[
h;1180;27;296]01201
[h;1306;92;297;4;1]207-555-2357[h;1647;103;298;5;1]25.00[y;1883;32;2055;2;H]
[s;2;244;1;299;2111;t;4;0;1]George[h;369;27;300]Smyth[h;499;133;301;6;1]1234
[h;711;25]Wash[h;820;23]Ave,[h;917;35]Peabody,[h;1109;32]MA[h;1181;26]01970
[h;1306;91;302;4;1]207-555-1212[h;1647;104;303;5;1]10.00[y;1883;33;2111;0;H]
[g;285;0;0;2150;2794,0]

Character Set

The characters in an XDOC file are encoded using the codes defined by Microsoft Windows code pages. XDOC markups embedded in the text use the ASCII (7 bits) subset of each code page. ScanSoft Inc's recognition software can recognize most characters included in the following Microsoft Windows code pages: Latin1 (1252), Central Europe (1250), Greek (1253), Turkish (1254), Cyrillic (1251) and Baltic (1257).

These Microsoft Windows code page definitions can be found at this web site:

<http://www.microsoft.com/globaldev/reference/wincp.asp>

The following characters are not recognized by ScanSoft Inc's recognition software (listed as hex values):

Latin1 (1252):

5e, 7e, 7f, 81, 83, 84, 85, 86, 87, 88, 89, 8b, 8d, 8e, 8f, 90, 96, 98, 99, 9b, 9c, 9e, a0, a2, a4, a6, a8, aa, ac, af, b2, b3, b4, b5, b7, b8, b9, ba, d7, de, fe

Western Europe (1250):

5e, 7e, 7f, 81, 83, 85, 86, 87, 88, 89, 8b, 90, 96, 98, 99, 9b, a0, a1, a2, a4, a6, a8, ac, b2, b4, b5, b7, b8, bd, d7, ff

Greek (1253):

Fe, 7e, 7f, 81, 83, 85, 86, 87, 88, 89, 90, 96, 98, 99, a0, a1, a4, a6, a8, aa, ac, af, b2, b3, b4, b5, d2, ff

Turkish (1254):

fe, 7e, 7f, 81, 83, 85, 86, 87, 88, 89, 8b, 8d, 8e, 8f, 90, 96, 98, 99, 9b, 9d, 9e, a0, a2, a4, a6, ac, af, b2, b3, b4, b5, b7, b8, b9, ba, d7

Cyrillic (1251):

5e, 7e, 7f, 85, 86, 87, 89, 8b, 96, 98, 99, 9b, a4, a6, ac, b5, b7

Baltic (1257):

5e, 7e, 7f, 81, 83, 85, 86, 87, 88, 89, 8a, 8b, 8c, 8d, 8e, 8f, 90, 96, 98, 99, 9a, 9b, 9c, 9d, 9e, 9f, a0, a1, a2, a4, a5, a6, a8, b2, b3, b4, b5, b7, b8, b9, d7, ff

ScanSoft also defines three control codes for representing three characters not represented in the supported code pages. The following characters are defined in the `icrpub.h` include file, and unlike the other characters, should always be referred to by their symbols rather than their values.

- **NOT_EQUAL:** This is the “not equal” (!=).
- **LESS_OR_EQUAL:** This is the “less than or equal” (<=).
- **GTR_OR_EQUAL:** This is the left “greater than or equal” (>=).

Index

A

Affine transformation, 4-, 9
Angular correction
 retrieving, 4-, 11

B

Boundaries
 zones, 4-, 5

C

Change font, 2-, 5, 10
Change region, 2-, 5, 10
Change subscript, 2-, 5
Change superscript, 2-, 5
Change underline, 2-, 5
Character confidence, 2-, 6, 14
Character operands, 2-, 2
Character set, 3-, 2
Compound documents, 1-, 1
Compound documents, 4-, 3
Concatenating files, 3-, 1
Confidence
 characters, 2-, 14
 word, 2-, 16
Confidence, 2-, 6

D

Defining regions, 4-, 4
Document name, 2-, 9, 10
Document name, 3-, 1
Documentation conventions., vi
Dropped capital letter, 2-, 5, 16

E

End cell table, 2-, 7
End escape character, 2-, 2
End of document, 2-, 9, 17
Essential modifiers
 page, 2-, 7
 text line, 2-, 5

F

Font description, 4-, 13

Font identifier, 2-, 8
Font information, 2-, 10
Font modifier
 page, 2-, 7
Fonts, 4-, 13
Frame, 4-, 5

G

Galleys, 4-, 4
Graphics formats, 1-, 1

H

Hard column break, 2-, 7, 16
Horizontal space, 2-, 5

I

Identifying fonts, 4-, 13
Image zone descriptor, 2-, 7, 16
Image zones, 4-, 7
ISO character set, 3-, 2

K

KDC_CCONF, 2-, 14
KDC_CHGFONT, 2-, 10
KDC_DOCNAME, 2-, 10
KDC_DROP, 2-, 16
KDC_FONTINFO, 2-, 10
KDC_HARD_COL, 2-, 16
KDC_HSPACE, 2-, 11, 12
KDC_LEADER, 2-, 13
KDC_LEXCL, 2-, 16
KDC_NDDOC, 2-, 17
KDC_NDLIN, 2-, 17
KDC_NDPAGE, 2-, 11
KDC_NDTABLE, 2-, 9
KDC_OHPHEN, 2-, 11
KDC_PZONE, 2-, 16
KDC_QABLE, 2-, 14
KDC_REGION, 2-, 10
KDC_RULE, 2-, 14
KDC_SCOL, 2-, 13
KDC_STABLE, 2-, 12
KDC_STDOC, 2-, 9
KDC_STLINE, 2-, 15
KDC_SUB, 2-, 10
KDC_SUPER, 2-, 15
KDC_TZONE, 2-, 15
KDC_UNDERLINE, 2-, 16
KDC_UNREC, 2-, 10
KDC_WBOX, 2-, 9
KDC_WCONF, 2-, 16

L

- Layout modifiers
 - page, 2-, 7
- Leader characters, 2-, 5, 13
- Lexical class, 2-, 6, 16
- Lexical modifiers
 - text line, 2-, 5
- Lexical order, 4-, 7

M

- Markups
 - syntax, 2-, 1
- Mode shift modifiers
 - text line, 2-, 5
- Modifier code, 2-, 2
- Modifier codes
 - alphabetical listing, 2-, 9, 10, 12, 13, 14, 15, 16
- Modifier end escape character, 2-, 2
- Modifier start escape character, 2-, 2
- Modifiers
 - alphabetical listing, 2-, 9
 - page, 2-, 7
 - text line, 2-, 4
- Modifiers, 2-, 2
- Multiple document, 3-, 1

N

- Nesting transactions, 2-, 4
- New table, 2-, 7, 12
- Nonprinting whitespace, 2-, 5, 11
- Numeric operands, 2-, 2

O

- Offset page image, 4-, 1
- Operand separator character, 2-, 2
- Operands
 - character, 2-, 2
 - numeric, 2-, 2
 - string, 2-, 2
- Operands, 2-, 2
- Optional hyphen, 2-, 6, 11

P

- Page image
 - zones, 4-, 3
- Page image analysis, 4-, 1
- Page image shear, 4-, 2
- Page image tilt, 4-, 2
- Page information, 2-, 7, 11
- Page modifiers, 2-, 7
- Page segmentation, 4-, 3
- Page segments
 - regions, 4-, 3

- zones, 4-, 3
- Page sequence, 3-, 1
- Page transactions
 - essential, 2-, 7
 - font modifier, 2-, 7
 - layout modifiers, 2-, 7
- Page transactions, 2-, 7
- Parsing XDOC Text, 3-, 3
- printing whitespace, 2-, 5
- Printing whitespace, 2-, 13

Q

- Questionable character, 2-, 6, 14

R

- Reading threads, 4-, 7
- Regions
 - defining, 4-, 4
- Regions, 4-, 3
- Retrieving angular correction, 4-, 11
- Ruling descriptor, 2-, 7, 14
- Rulings
 - transformations, A-, 4
- Rulings, 4-, 7
- Running text vs. forms, 4-, 4

S

- Sample page image, B-, 1
- Sample XDOC file, 2-, 1
- Sample XDOC Text, B-, 2
- Scale
 - text vs. image, 4-, 3
- Section break, 2-, 7
- Section change, 2-, 12
- Separator character, 2-, 2
- Shift to/from subscript, 2-, 10
- Shift to/from superscript, 2-, 15
- Shift to/from underline, 2-, 16
- Skewed page image, 4-, 1
- Start escape character, 2-, 2
- Start new table row, 2-, 13
- Start of document, 2-, 9
- Start table cell, 2-, 13
- Start table column, 2-, 7
- Start-of-page, 2-, 7, 14
- Start-of-text line, 2-, 5, 15
- String operands, 2-, 2

T

- Technical Specification
 - documentation conventions,, vi
 - organization,, v
- Technical support,, vi

- Terminating transactions, 2-, 3
- Text data, 3-, 2
- Text line information, 2-, 5, 17
- Text line modifiers, 2-, 4
- Text line transactions
 - essential, 2-, 5
 - mode shift modifiers, 2-, 5
- Text line transactions, 2-, 4
- Text line transformations, A-, 1
- Text parser, 3-, 3
- Text transaction syntax, 2-, 9
- Text zone descriptor, 2-, 7, 15
- ScanSoft SDK Programmer's Guide., vi
- ScanSoft SDK., v
- Textline transactions,
 - lexical modifiers, 2-, 5
- Tilt
 - page image, 4-, 2
- Topological layout, 4-, 5
- Transactions
 - nesting, 2-, 4
- Transactions, 2-, 3
- Transformations
 - algebraic, 4-, 9
 - image coordinates to page coordinates, A-, 1
 - page to image, 4-, 9
 - rulings, 4-, 7
 - rulings, A-, 4
 - text line, A-, 1
 - text, 4-, 2
 - zones, 4-, 6
 - zones, A-, 3

U

- Unrecognized character, 2-, 6, 10

V

- Viewing XDOC files, 3-, 2

W

- Word bounding box, 2-, 6
- Word bounding box. 2-, 9
- Word bounding boxes
 - coordinates, 4-, 11
- Word confidence, 2-, 6, 16

X

- XDOC
 - flexible grammar, 1-, 1
 - markups, 2-, 1
 - transactions, 2-, 3
- XDOC coordinate system, 4-, 1
- XDOC define, 2-, 9
- XDOC features, 1-, 1

- XDOC file hierarchy, 1-, 1
- XDOC images, 1-, 1
- XDOC Text
 - file structure, 3-, 1
- XDOC Text file
 - concatenating, 3-, 1
 - multiple documents, 3-, 1
 - page sequence, 3-, 1
- XDOC Text markups, 2-, 9
- XDOC Text pages
 - running text, 4-, 4
- XDOC Text, 1-, 1

Z

- Zone
 - frame, 4-, 5
- Zones
 - boundaries, 4-, 5
- Zones, 4-, 3