



# Practical Formatting Using XSL-FO (Extensible Stylesheet Language Formatting Objects)

Crane Softwrights Ltd.  
<http://www.CraneSoftwrights.com>





# Practical Formatting Using XSL-FO (Extensible Stylesheet Language Formatting Objects)

Crane Softwrights Ltd.  
<http://www.CraneSoftwrights.com>

## Copyrights

- Pursuant to <http://www.w3.org/Consortium/Legal/ipr-notice.html>, some information included in this publication is from copyrighted material from the World Wide Web Consortium as described in <http://www.w3.org/Consortium/Legal/copyright-documents.html>: Copyright (C) 1995-2008 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. The status and titles of the documents referenced are listed in the body of this work where first used.
- Other original material herein is copyright (C) 1998-2008 Crane Softwrights Ltd. This is commercial material and may not be copied or distributed by any means whatsoever without the expressed permission of Crane Softwrights Ltd.

## Disclaimer

- By purchasing and/or using any product from Crane Softwrights Ltd. ("Crane"), the product user ("reader") understands that this product may contain errors and/or other inaccuracies that may result in a failure to use the product itself or other software claiming to utilize any proposed or finalized standards or recommendations referenced therein. Consequently, it is provided "AS IS" and Crane disclaims any warranty, conditions, or liability obligations to the reader of any kind. The reader understands and agrees that Crane does not make any express, implied, or statutory warranty or condition of any kind for the product including, but not limited to, any warranty or condition with regard to satisfactory quality, merchantable quality, merchantability or fitness for any particular purpose, or such arising by law, statute, usage of trade, course of dealing or otherwise. In no event will Crane be liable for (a) punitive or aggravated damages; (b) any direct or indirect damages, including any lost profits, lost savings, damaged data or other commercial or economic loss, or any other incidental or consequential damages even if Crane or any of its representatives have been advised of the possibility of such damages or they are foreseeable; or (c) for any claim of any kind by any other party. Reader acknowledges and agrees that they bear the entire risk as to the quality of the product.

# Practical Formatting Using XSL-FO (Prelude) (cont.)



---

## Preface

The main content of this book is in an unconventional style primarily in bulleted form

- derivations of the book are used for instructor-led training, requiring the succinct presentation
  - note the exercises included in instructor-led training sessions are not included in the book
- derivations of the book can be licensed and branded for customer use in delivering training
- the objective of this style is to convey the essence and details desired in a compact, easily perused form, thereby reducing the search for key words and phrases in lengthy paragraphs
- each chapter of the book corresponds to a module of the training
- each page of the book corresponds to a frame presented in the training
- a summary of subsections and their pages is at the back of the book

Much of the content is hyperlinked both internally and externally to the book in the 1-up full-page sized electronic renditions:

- note when using the Acrobat Reader for navigation, the history "back" keystroke sequence is "Ctrl-Left"
- page references, e.g.: Chapter 3 Basic concepts of XSL-FO (page 44)
  - the back-of the book index is hyperlinked to the body of the book
  - the letter references at the bottom of each page are hyperlinked to the index
- construct references are typeset with conventions
  - formatting objects are in monospaced text in brackets, e.g.: `<basic-link>`
  - properties are followed by "=", e.g.: `baseline-shift=`
  - data types are in proportional text in brackets, e.g.: `<angle>`
- references to sections of the Recommendation are in parentheses, e.g.: (7.13.3)
- external references are in monospaced text, e.g.:  
`http://www.w3.org/TR/2001/REC-xsl-20011015/xslspec.html`
- chapter references in book summary
- section references in chapter summary
- subsection references in table of contents at the back of the book
- no hyperlinks are present in the cut, stacked, half-page, or 2-up renditions of the material

# Practical Formatting Using XSL-FO



- Introduction - Paginating structured information
- Chapter 1 - Introducing XSL-FO
- Chapter 2 - The context of XSL-FO
- Chapter 3 - Basic concepts of XSL-FO
- Chapter 4 - Area and page basics
- Chapter 5 - Generic body constructs
- Chapter 6 - Tables
- Chapter 7 - Floats, footnotes and containers
- Chapter 8 - Flows, static content and page geometry sequencing
- Chapter 9 - Bookmarks and indexes
- Chapter 10 - Breaks, keeps, spacing, borders and backgrounds
- Chapter 11 - Supplemental objects
- Chapter 12 - Interactive objects
- Chapter 13 - Where XSL-FO 1 falls short
- Annex A - Using XSLT with XSL-FO
- Annex B - XSL-FO expressions
- Annex C - XSL-FO object summary
- Annex D - XSL-FO property summaries
- Annex E - Sample tool information
- Conclusion - Where To Go From Here?

Series: Practical Formatting Using XSL-FO

Reference: PFUX

Pre-requisites:

- knowledge of XML syntax

Outcomes:

- exposure to example scripts
- exposure to basic terminology
- exposure to every formatting object

# Paginating structured information

Introduction - Practical Formatting Using XSL-FO



This book is oriented to the XSL-FO stylesheet writer, not the XSL-FO processor implementer

- certain behaviors important to an implementer are not included
- objective to help a stylesheet writer understand the language facilities needed to solve their problem
  - a language reference arranged thematically to assist comprehension
  - a different arrangement than found in the Recommendation itself

This book covers every formatting object of XSL-FO both versions 1.0 and 1.1:

- ¶ content specific to XSL-FO 1.0 is marked with a "1.0" icon at the beginning of the line
- ¶ content specific to XSL-FO 1.1 is marked with a "1.1" icon at the beginning of the line

First two chapters are introductory in nature

- overview of context of XSL-FO amongst other members of the XML family of Recommendations
- basic flow diagrams illustrate use of XSL-FO
- basic terminology and concepts are defined and explained

Third chapter covers the basics of the area model and page model

- an understanding of the conceptual rendering areas being created by the stylesheet writer
- important to understand these models in order to apply the language features

Fourth through twelfth chapters address XSL-FO vocabulary

- all objects are described in detail and their properties are summarized
- significant or important properties are highlighted and described

Thirteenth chapter outlines where the XSL-FO vocabulary falls short

- formatting requirements that are often needed but not available

## Paginating structured information (cont.)

Introduction - Practical Formatting Using XSL-FO



---

### Introduction (cont.)

First annex overviews issues of XSLT when working with XSL-FO

- dependencies on XSLT by XSL-FO
- issues to remember when writing XSLT

Second annex covers XSL-FO expressions

- rules when writing expressions
- issues to remember

Third and fourth annexes include object and property summaries derived from the Recommendation

- groupings of objects and properties
- alphabetical lists of objects and properties
- print-oriented summary of all productions

Last annex addresses questions regarding tools

- lists of questions for processor implementers when assessing tool capabilities

External ZIP file included with the purchase of the book

- all of the complete scripts utilized in the documentation as stand-alone files ready for analysis and/or modification

# Chapter 1 - Introducing XSL-FO



- 
- Introduction - Contrasting browsed vs. paginated presentations
  - Section 1 - Fine-grained control of layout nuances
  - Section 2 - The many existing deployments of XSL-FO

PREVIEW EXCERPT



# Contrasting browsed vs. paginated presentations

Chapter 1 - Introducing XSL-FO



Consider two media for presenting information to users

- presenting information in a browser window is fundamentally different than presenting the same information in printed form
- browser screens are dynamic
  - navigating around information is the responsibility of the browser
    - in response to user actions
  - the page dimensions can be changed by the reader of the information
    - the width can be modified after the information is presented, requiring the information to be reflowed in the new dimensions
    - the length of the screen is essentially infinite and grows to accommodate the amount of information on the page
- printed pages are static
  - navigating around information is the responsibility of the reader
    - based on cues given in the printed text
  - the page dimensions are fixed by the presenter of the information

Maintaining information separately for two presentations is troublesome

- perhaps the HTML presentation is authored separately from the printed presentation
  - since the navigation tools are different, each authoring tool gives the author a different user interface and authoring options
- the lack of a common source of content introduces maintenance challenges
  - problems when ensuring changes in one are reflected in the other
- best to have a single source of information to rearrange and re-label for different presentations

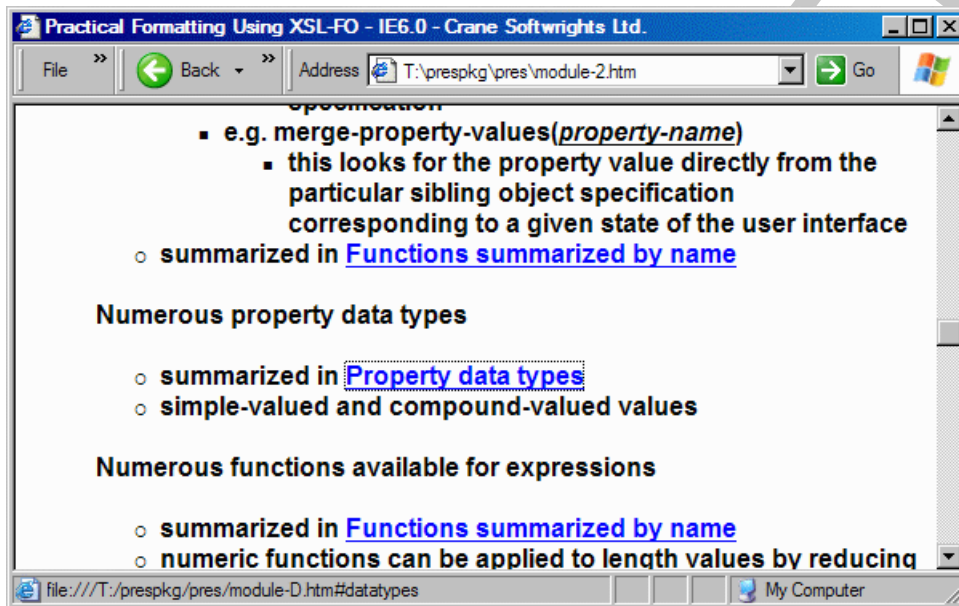
# Contrasting browsed vs. paginated presentations (cont.)

Chapter 1 - Introducing XSL-FO



Consider a fragment of training content presented in a browser screen

- hyperlinks are recognized by the underscored text in blue
  - in a black and white print this may be confused with the underscored text at the top of the window that isn't a hyperlink
- a hyperlink with focus exposes the target of the hyperlink in the status line of the window
  - requires interaction with the browser to check each of the possible target locations
- traversing the hyperlink involves interacting with the user agent





# Contrasting browsed vs. paginated presentations (cont.)

Chapter 1 - Introducing XSL-FO

Consider the training information being authored in HTML for the web presentation:

```
01 <p>Numerous property data types</p>
02 <ul>
03   <li>summarized in <a href="module-D.htm#datatypes">Property
04   data types</a></li>
05   <li>simple-valued and compound-valued values</li>
06 </ul>
07 <p>Numerous functions available for expressions</p>
08 <ul>
09   ...
10 <h2><a name="datatypes">Property data types</a></h2>
```

Of note:

- lines 3-4 show the markup for the highlighted hyperlink
  - the title of the referenced item in line 10 is copied as the "clickable text" of the canvas
  - changing the title requires changing all references to the title
- the entire web page could span dozens of printed pages
  - the hyperlink information is lost
    - no indication of where a target of a hyperlink is found
  - no tools for navigating the collection of pages

Authoring the information in a word processor would require similar duplication of information

- changes to a title requiring changes to all references to the title
- adding of page numbers and page number citations
  - necessitating distinct content from the web-based content

# Contrasting browsed vs. paginated presentations (cont.)

Chapter 1 - Introducing XSL-FO



Consider an XML representation of the same training material

- we can use any vocabulary of element types and attributes to represent the concepts of training information
- the hyperlink to another location can be an empty element
  - all that is needed is to know what is being referenced not how it is presented

```

01 <course>
02   <title>Practical Formatting Using XSL-FO</title>
03 ...
04 <module id="basic">
05   <title>Basic concepts of XSL-FO</title>
06   <lesson id="vocab">
07     <title>Formatting object XML vocabulary</title>
08     <frame id="propexp">
09       <title>Property value expressions</title>
10 ...
11   <point>summarized in <ref idref="funcname"/></point>
12 </points>
13 <para>Numerous property data types</para>
14 <points>
15   <point>summarized in <ref idref="datatypes"/></point>
16   <point>simple-valued and compound-valued values</point>
17 </points>
18 <para>Numerous functions available for expressions</para>
19 <points>
20   <point>summarized in <ref idref="funcname"/></point>
21 ...
22   </frame>
23 </lesson>
24 </module>
25 ...
26   <frame id="funcname">
27     <title>Functions summarized by name</title>
28 ...
29   </frame>
30 ...
31   <frame id="datatypes">
32     <title>Property data types</title>
33 ...
34   </frame>
35 ...
36 </course>

```



## Contrasting browsed vs. paginated presentations (cont.)

Chapter 1 - Introducing XSL-FO

---

Of note:

- the same hyperlink is an empty XML element on line 15
  - attribute points to the frame being referenced
- transformation can obtain the title of the referenced section at presentation time
  - only one place to maintain the title
  - guarantees consistency in presentation during maintenance

The Extensible Stylesheet Language Transformations (XSLT) is used for presentation and transformation

- rearranges instances of XML information into instances of other vocabularies
- e.g. an XSLT tool reads our XML instances and produces an instance of the HTML vocabulary
  - the stylesheet dictates how the HTML is created with the hyperlink reference for traversal and the clickable content for the canvas

# Contrasting browsed vs. paginated presentations (cont.)

## Chapter 1 - Introducing XSL-FO



Consider the same training information presented in the printed form


- basic presentation of the material is similar to that of the browser presentation
- content is paginated over a number of fixed-size pages
- navigation tools are different for the collection of pages

*Practical Formatting Using XSL-FO*

**Property value expressions**

Chapter 2 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



---

A property's value can be the evaluation of an expression

- may include fixed values
  - e.g. `space-before="20pt div 2"`
- may include contextually-sensitive values
  - e.g. `space-before="from-parent(font-size) div 2"`
- same operators as in XPath 1.0
- includes same operands as in XPath 1.0
- includes length values as operands that are not allowed in XPath 1.0
- expressions influenced by the font size evaluate the font size before evaluating any other components of the expression

Core function library defined for property expressions

- functions with access to property values of the current node
  - e.g. `inherited-property-value(property-name)`
    - this obtains a value that may be specified on the current node or may be inherited from the closest ancestral node that specifies the value
- functions with access to property values of other nodes
  - e.g. `from-parent(property-name)`
    - this looks for the property value directly from the parent object specification
  - e.g. `merge-property-values(property-name)`
    - this looks for the property value directly from the particular sibling object specification corresponding to a given state of the user interface
- summarized in Functions summarized by name (page 307)

Numerous property data types

- summarized in Property data types (page 327)
- simple-valued and compound-valued values

Numerous functions available for expressions

- summarized in Functions summarized by name (page 307)
- numeric functions can be applied to length values by reducing the "unit power" and adding it back again after
  - a length has a unit power of 1, while a number has a unit value of zero
  - e.g. `round()` takes a number argument and not a length argument
  - can use: `round( length-value div 1.0cm ) * 1.0cm`

Copyright © Crane Softwrights Ltd.

Third Edition - 2002-09-05 - ISBN 1-894049-10-1  
Information subject to restrictive legend on first page.

Page 41 of 405



# Contrasting browsed vs. paginated presentations (cont.)

Chapter 1 - Introducing XSL-FO

Of note:

- the hyperlinks are rendered with the page number of the target of each reference
- the page number is shown at the bottom right of the page for navigation
- the module and lesson information is shown at the top left for context

The Extensible Stylesheet Language Formatting Objects (XSL-FO) XML vocabulary is used to express pagination semantics

- architecturally the same as when we use HTML to express browser semantics
- we learn the XSL-FO vocabulary representing the XSL-FO semantics
  - e.g. how to express a hyperlink as a page number citation
  - same as we needed to know what HTML can do in a browser
    - e.g. how to express a hyperlink as an anchor
- instances of our XML vocabularies are transformed into instances of the XSL-FO vocabulary
  - e.g. transforming our `<ref/>` elements into `<page-number-citation/>` elements
  - same process as is needed to go from XML to HTML
    - e.g. transforming our `<ref/>` elements into `<a>` elements
- an XSL-FO engine interprets the XSL-FO semantics to produce the printable results
  - e.g. an XSL-FO tool reads our XSL-FO and produces a PDF file
  - same process as is needed by a web browser for HTML
    - e.g. a browser user agent reads our HTML and renders a screen window
- only difference is the selection of presentation semantics appropriate for the desired target medium

XSL-FO offers many nuances of flow and formatting semantics

- fine-grained control over layout and appearance

Audiences for browser windows can be different than those for printed information

- a whole constituency of users does not accept reading information from a computer screen
  - will only read information from printed pages
- those who do read lengthy documents from browser windows periodically need a printed rendition
  - the navigation tools on the screen are not translated to the paper form in a browser print request
- often the different perspective requires a different arrangement
  - not just a decorated version of the same arrangement of the information

# Fine-grained control of layout nuances

Chapter 1 - Introducing XSL-FO

Section 1 - Fine-grained control of layout nuances



XSL-FO gives control to the stylesheet writer over many layout constructs useful in printing applications

- pagination
  - static content (headers and footers)
  - page sequences (odd/even, first/middle/last, blank/not-blank)
- blocks of lines and inline areas
- lists and parallel blocks
- images (external and embedded)
- hyperlinks (internal and external)
- leaders (elastic and inelastic)
- floats, footnotes, containers
- tables
- change bars
- bookmarks
- indexes
- bi-directional text protection





# The many existing deployments of XSL-FO

Chapter 1 - Introducing XSL-FO

Section 2 - The many existing deployments of XSL-FO

Vendors report wide-spread deployment of this technology in real-world high-volume publishing requirements

- what follows is only a sample list of applications
- check vendor sites for case studies

Day-to-day life touches on documents meeting stringent layout requirements

- presentation and training material handouts
- high-school student study guides
- online train tickets in Germany
- account statements and telephone bills
- technical manuals, user guides and maintenance documentation
  - e.g. Daimler and Nokia
- share portfolio overviews and 401K statements
- printed box forms for manual entry
- multilingual pension letters
- bulk mail applications
- mortgage documents
- commercial books
  - e.g. Prentice Hall and O'Reilly Safari

Government use:

- U.S. intelligence documents
  - publishing in XSL-FO with a copy from the XSL-FO into HTML
- legislation (bills, acts and amendments)
- "The Daily Threat Assessment" (U.S.)
  - varying detail and classification levels; distribution to the Joint Chiefs of Staff
- government forms
- XBRL reports for Federal Financial Examination Council
- Medicare-related applications and enrollment booklets

Public use:

- digitally-signed XSL-FO stylesheets
  - allows a commercial-grade tool to be distributed freely for only signed stylesheets
    - any change to the stylesheet prevents the processor from using it
  - protects vendor from a customer using a publicly-available tool for private use
- printing UBL invoices using the United Nations Layout Key
  - see Crane's web site for an example

## Chapter 2 - The context of XSL-FO



- 
- Introduction - Overview
  - Section 1 - The XML family of Recommendations
  - Section 2 - Examples

### Outcomes:

- introduction to objectives and purpose
- awareness of available documentation
- an awareness of available documentation and a small subset of publicly available resources
- exposure to example scripts

# Overview

## Chapter 2 - The context of XSL-FO



In this chapter we look at four technologies. In later chapters we will both set the stage for and delve into the semantics of paginated formatting available through detailed examples, definitions of terminology, and an examination of each of the formatting objects.

### Extensible Markup Language (XML)

- hierarchically describes an instance of information
  - using embedded markup according to rules specified in the Recommendation
  - according to a vocabulary (a set of element types each with a name, a structure and optionally some attributes) described by the user
- optionally specifies a mechanism for the formal definition of a vocabulary
  - controls the instantiation of new information
  - validates existing information

### Document Style Semantics and Specification Language (DSSSL)

- an internationally standardized collection of style semantics
- a specification language for the transformation of structured information and the application of the standardized internationalized style semantics for paginating structured information

### Cascading Stylesheets (CSS)

- a set of formatting properties to attach to structured documents
  - defines a cascade of property sources from the markup to the embedded stylesheet to the external stylesheet to the default presentation semantics of the processor
- browsers supporting CSS can render the document structure of HTML documents or XML documents according to the properties

### Extensible Stylesheet Language Family (XSLT/XSL/XSL-FO)

- XSL Transformations (XSLT)
  - specifies the transformation of XML-encoded information into a hierarchy using the same or a different document model *primarily for the kinds of transformations for use with XSL*
- XSL (Formatting Semantics, a.k.a. XSL-FO)
  - specifies the vocabulary and semantics of the formatting of information for paginated presentation
  - colloquially referred to at times as XSL Formatting Objects

# Extensible Markup Language (XML)

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



- 
- <http://www.w3.org/TR/REC-xml>

A Recommendation fulfilling two objectives for information representation

- capturing information in a hierarchical form in markup according to basic XML-defined constraints
  - creating well-formed documents of elements, attributes and other constructs
- restricting and/or validating hierarchical information in XML to arbitrary user-specified constraints
  - defining a model or grammar for the structure and content of a document
    - collection of available element types and their respective attributes
      - inherent relationships between information in the hierarchy
    - vocabulary can be expressed formally in XML 1.0 as a Document Type Definition (DTD)

Nothing in XML is related to presentation or rendition

- no inferred semantics for the display or formatting of information when using XML
- `xml:space` is used only for the significance of the white space in the document

The vocabulary of elements and attributes used in an instance can be validated

- at a grammar level by a declarative document model
  - structural validation
    - the nesting and order of elements and their use of attributes
  - lexical validation and integrity
    - certain aspects of content and the allowable string values of attributes
  - a distinct process separate from the applications acting on the information
    - analysis against the DTD
    - analysis using other validation mechanisms (e.g. XML Schema, RELAX-NG Schema, Schematron, etc.)
- at a semantic level by the application processing the information
  - the semantics of information is not defined by the grammar or structure of the information
  - information "means" exactly what any application processing the information wants it to mean
  - the application analyzes the structure and content of the information for appropriateness to its purpose
    - can test conditions or constraints that cannot be expressed in a formal document model syntax
    - can algorithmically determine validity to support requirements not easily expressed declaratively



# Extensible Markup Language (XML) (cont.)

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations

XML vocabularies can be translated to an application's specific vocabulary

- our XML vocabularies should be designed according to the business processes acting on the information, not around the appearance
  - flexibility to have many different appearances for the same information
- a presentation application vocabulary could just be attribute values added to our own vocabularies without changing element names (e.g. Cascading Stylesheets (CSS))
  - we can add properties that are recognized by a browser
  - our information is then rendered visually or aurally according to the properties
- a presentation application vocabulary could be elements and attributes expressing the semantics of browsing information (e.g. Hypertext Markup Language (HTML) or Scalable Vector Graphics (SVG))
  - browsers have adopted common presentation semantics for HTML constructs
  - if the presentation semantics are sufficient, we need only use HTML
  - if the presentation semantics are insufficient, we can use both HTML and CSS
    - CSS-aware browsers would present the information as desired
    - non-CSS-aware browsers would at least present the information using the presentation semantics of HTML

Namespaces distinguish constructs in information from different vocabularies

- a single instance can contain information from different document models
- the recognition of element types by an application is through combination of namespace URI string and un-prefixed element type name
  - the prefix used in the instance is irrelevant to the application

An application can only act on the vocabulary it recognizes and must have a behavior for vocabulary it doesn't recognize

- a web browser understands the HTML and CSS vocabularies
  - displays understood constructs accordingly
  - ignores unrecognized constructs while passing content through to the canvas
- an e-commerce application understands the vocabulary designed to trigger behavior
  - performs the functionality accordingly
  - could exit with an error or warning for unrecognized constructs
- a namespace-aware application can recognize constructs from different vocabularies
  - using namespaces gives us better labels for the elements and attributes in our XML information than simple names
  - the semantics of our information is assumed by the application we use to process our information through the labels we use
    - using better labels results in more successful application processing

## XML information links

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



### Links to useful information

- <http://www.xml.com/axml/axml.html> - annotated version of XML 1.0
- <http://xml.coverpages.org/xml.html> - Robin Cover's famous resource collection
- <http://xml.coverpages.org/xll.html> - Extensible Linking Language
- <http://xml.silmaril.ie/> - Peter Flynn FAQ
- <http://www.xmlbooks.com/> - a summary of available printed books
- <http://www.CraneSoftwrights.com/links/trn-20080127.htm> - training material
- <http://www.CraneSoftwrights.com/resources> - free resources
- <http://XMLGuild.info> - consulting and training expertise
- <http://xml.coverpages.org/elementsAndAttrs.html> - a summary of opinions

### Related initiatives and specifications

- <http://www.w3.org/TR/2004/REC-xml-infoset-20040204> - XML Information Set
- <http://www.w3.org/TR/xmlschema-0/> - W3C XML Schema
- <http://www.relax-ng.org> - ISO/IEC 19757-2 RELAX NG (based on RELAX and TREX)
- <http://www.schematron.com> - ISO/IEC 19757-3 Schematron
- <http://www.nvdl.org> - ISO/IEC 19757-4 Namespace-based Validation Dispatching Language (NVDL)
- <http://www.w3.org/TR/DOM-Level-2/> - Document Object Model Level 2
- <http://www.saxproject.org> - Simple API for XML



# Document Style Semantics and Specification Language (DSSSL)

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations

---

- ISO/IEC-10179:1996
  - <http://www.y12.doe.gov/sgml/wg8/dsssl/readme.htm>

Transforming and formatting structured information

- distinguishes the separate behaviors required to style structured information
- a transformation language to rearrange structured information
- pagination semantics for presenting information in fixed-sized folios
- includes an extension mechanism for arbitrary formatting semantics

A programming language for transforming structured information

- specifies relationships between multiple input documents in the transformation to zero or more output documents
- uses side-effect-free dialect of Scheme (itself a derivative of LISP) for the expression language

A standardized set of formatting semantics for paginated output

- specifies the intent of the result of a formatting process
- does not specify the rendering process
- no bias in constructs to any particular writing direction

A framework for implementation-defined sets of formatting semantics

- there exists a set of semantics for formatting into SGML/XML markup, thus implementing a transformation function through the use of formatting facilities

Custody of ISO/IEC JTC 1/SC 34/WG 2

- formerly ISO/IEC JTC 1/WG 4
- formerly ISO/IEC JTC 1/SC 18/WG 8



# Cascading Stylesheets (CSS)

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



- 
- <http://www.w3.org/TR/REC-CSS1>
  - <http://www.w3.org/TR/REC-CSS2>

## Formatting property assignment for web documents (HTML and XML)

- no document manipulation capabilities
- width and length of presentation are not fixed
  - can be changed dynamically by the reader of the information
- developed to address incompatible vendor extensions for formatting that were being added to browsers

## Ornamentation of the document tree

- attaching stylistic information to nodes
- simple prefixing and suffixing of nodes with text
- control of white space around information
- overlapping and transparent rectangular regions
- significant use of inheritance of formatting properties from ancestral tree locations
  - defines the "cascade" of application of inheritable formatting properties

## Multiple media type support

- character display presentation properties
- tabular presentation properties
- aural presentation properties for visually impaired browsing
  - disabled users
  - mobile users

## Doesn't (*shouldn't*) interfere with legacy browsers not supporting CSS

- values expressed in attributes and document metadata
- expression can be external to the document itself
  - required to be external for XML files not using namespaces
- can be introduced to namespace-aware XML documents through the HTML vocabulary

## Working group is producing a common formatting model for web documents

- all W3C Recommendations needing presentation properties should use the CSS semantics and associated property names where applicable





# Styling structured information

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations

Styling is *transforming* and *formatting* information

- the application of two processes to information to create a rendered result
- the ordering of information for creation isn't necessarily (or shouldn't be constrained to) the ordering of information for presentation or other downstream processes
  - it is a common (though misdirected) first step for people working with these technologies to focus on presentation
  - the ordering should be based on business rules and inherent information properties, not on artificial presentation requirements
  - downstream arrangements can be derived from constraints imposed upstream in the process
  - information created richly upstream can be manipulated into less-richly distinguished information downstream, but not easily the other way around
  - exception when the business rules are presentation or appearance oriented (e.g. book publishing)
- the need to present information in more than one arrangement requires transformation
- the need to present information in more than one appearance requires formatting

W3C XSL Working Group

- chartered to define a style specification language that covers at least the formatting functionality of both CSS and DSSSL
- not intended to replace CSS, but to provide functionality beyond that defined by CSS
  - e.g. add element reordering and pagination semantics

Two W3C Recommendations

- designed to work together to fulfill these two objectives
- XSL Transformations (XSLT) - versions 1.0 and 2.0
  - transforming information obtained from a source into a particular reorganization of that information to be used as a result
- Extensible Stylesheet Language (XSL/XSL-FO) - versions 1.0 and 1.1
  - specifying and interpreting formatting semantics for the rendering of paginated information
  - the acronym XSL-FO is unofficial but in wide use, including at the W3C, for just the formatting objects, properties and property values
  - XSL normatively includes XSLT by reference in chapter 2
    - XSLT has specific features designed to be used for XSL-FO

XSLT and XSL-FO are endorsed by members of WSSSL

- an association of researchers and developers passionate about markup technologies

# Extensible Stylesheet Language Transformations (XSLT)

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



- ① <http://www.w3.org/TR/xslt>
- ② <http://www.w3.org/TR/xslt20>

## Transformation using construction by example

- a vocabulary for specifying templates of the result that are filled-in with information from the source
  - the stylesheet includes examples of each of the components of the result
  - the stylesheet writer declares how the XSLT processor builds the result from the supplied examples
- the primary memory management and manipulation (node traversal and node creation) is handled by the XSLT processor using declarative constructs, in contrast to a transformation programming language or interface (e.g. the DOM - Document Object Model) where the programmer is responsible for handling low-level manipulation using imperative constructs
- includes constructs to iterate over structures and information found in the source
- the information being transformed can be traversed in different ways any number of times required to construct the desired result
- straightforward problems are solved in straightforward ways without needing to know programming
  - useful, commonly-required facilities are implemented by the processor and can be triggered by the stylesheet
  - the language is Turing complete, thus arbitrarily complex algorithms can be implemented (though not necessarily in a pretty fashion)
- includes constructs to manage stylesheets by sharing components in different fragments
- ③ XSLT 2.0 has many more programming features and function calls than XSLT 1.0

## Not intended for syntactic general purpose XML transformations

- designed for downstream-processing transformations suited for use with XSL formatting vocabulary
  - includes facilities for working with the XSL vocabulary easily
- still powerful enough for *most* downstream-processing transformation needs
  - an XSLT stylesheet can be (and is) called a transformation script
  - absolutely general purpose when the output from XSLT is going to be input to an XML processor
- does not include certain features appropriate for syntax-level general purpose transformations
  - unsuitable for original markup syntax preservation requirements
- ③ XSLT 2.0 has many more syntax serialization features than XSLT 1.0

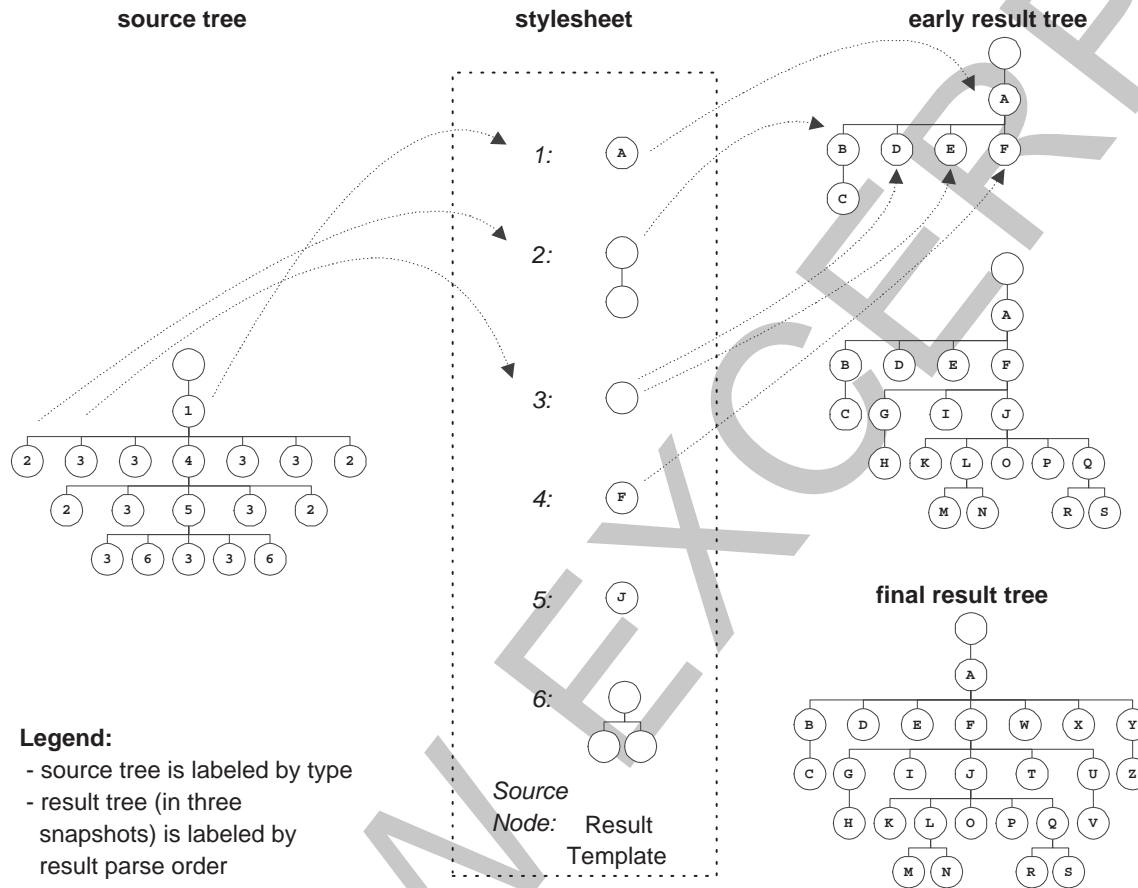


# Extensible Stylesheet Language Transformations (XSLT) (cont.)

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations

Illustration of triggered templates constructing a result:



Of note:

- the source tree contains nodes of six different types, labeled "1" through "6"
  - a number of nodes are found multiple times in the source tree
- the stylesheet contains fragmented examples of the result tree
  - each example template is associated with a node in the source tree
- the nodes in the source tree trigger the building of the result from the example templates
  - some examples are used multiple times in the result
- in this example, the source tree is visited strictly in parse order to generate the result tree
  - the stylesheet can visit the source tree in whatever order is required to trigger the assembly of the result tree in result parse order
  - result parse order is indicated by the letters "A" through "Z"
- the node at the very top of the source tree and the result tree is a root node that does not represent any actual information

# Extensible Stylesheet Language (XSL/XSL-FO)

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



- 
- ① <http://www.w3.org/TR/2001/REC-xsl-20011015/>
  - ① <http://www.w3.org/TR/xsl11> (<http://www.w3.org/TR/xsl>)

## Paginated flow and formatting semantics vocabulary

- capturing agreed-upon formatting semantics for rendering information in a paginated form on different types of media
- XSLT is normatively referenced as an integral component of XSL as a language to transform an instance of an arbitrary vocabulary into the XSL-FO XML vocabulary
- XSL-FO can be regarded simply as a "pagination markup language"
- flow semantics from the DSSSL heritage
  - e.g. headers, footers, page numbers, page number citations, columns, etc.
- formatting semantics from the CSS heritage
  - e.g. visual properties (font, color, etc.) and aural properties (speak, volume, etc.)

## Target of transformation

- the stylesheet writer transforms a source document into a hierarchy that uses only the formatting vocabulary in the result tree
- stylesheet is responsible for constructing the result tree that expresses the desired rendering of the information found in the source tree
  - the XML document gets transformed into its appearance
- stylesheet cannot use any user constructs as they would not be recognized by an XSL rendering processor
  - for example, the rendering engine doesn't know what an invoice number or customer number is that may be represented in the source XML
  - the rendering engine does know what a block of text is and what properties of the block can be manipulated for appearance's sake
  - the stylesheet transforms the invoice number and customer number into two blocks of text with specified spacing, font metrics, and area geometry

## Device-independent formatting constructs

- the XSL-FO vocabulary describes two media interpretations for objects and properties:
  - visual media
  - aural media
  - a further distinction is also made at times for interactive media
- the results of applying a single stylesheet can be rendered on different types of rendering devices, e.g.: print, display, audio, etc.
- may still be appropriate to have separate stylesheets for dissimilar media
  - device independence allows the information to be rendered on different media, but a given rendering may not be conducive to consumption

# Styling semantics and vocabularies

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



XSLT and XSL-FO processors implement styling semantics

- recognize standardized constructs by their labels in the two respective namespaces
  - elements and their attributes represent semantic concepts
    - XSLT instructions and their controls
    - XSL-FO formatting objects and their properties
- recognize extension constructs by their labels in namespaces recognized by the processor
- accommodate constructs by their labels in unrecognized namespaces

XSLT and XSL-FO document type definitions are described using prose

- there are no standardized XML 1.0 DTD representations of the grammar of the vocabularies
  - DTD semantics and syntax unable to fully express all of the grammatical constraints
- XSLT 1.0 and XSL 1.0 Recommendations describe the document type definitions
- processors do all aspects of validation and interpretation according to the respective document type
- snippets of DTD content model syntax with Kleene operators used in documentation because of the familiarity with the reader
  - "?" for zero or one
  - "\*" for zero or more
  - "+" for one or more
- additional constraints not expressible in DTD content model syntax are described in prose

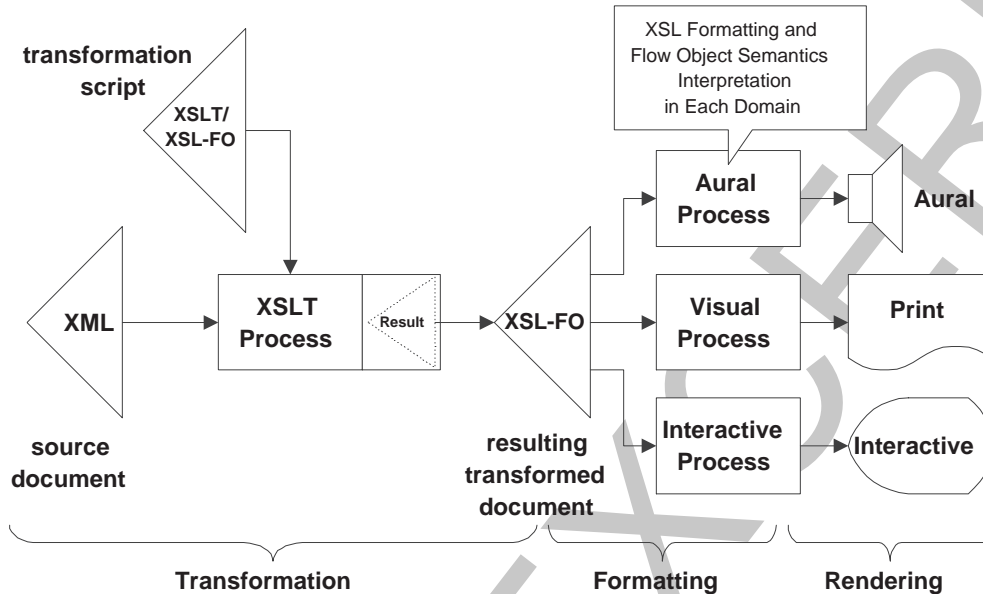
# Outboard XSLT and XSL-FO processes

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



The XSL-FO and foreign object vocabularies can be used in a standalone XML instance, perhaps as the result of an XSLT transformation using an outboard XSLT processor:



Note the same three distinct phases as when XSLT and XSL-FO processors are combined in a single application:

- transformation creates XSL-FO expressing our intent for formatting the source XML
- XSL-FO process interprets our intent into the information that is to be rendered on the target device
- XSL-FO process effects the rendering to reify the result





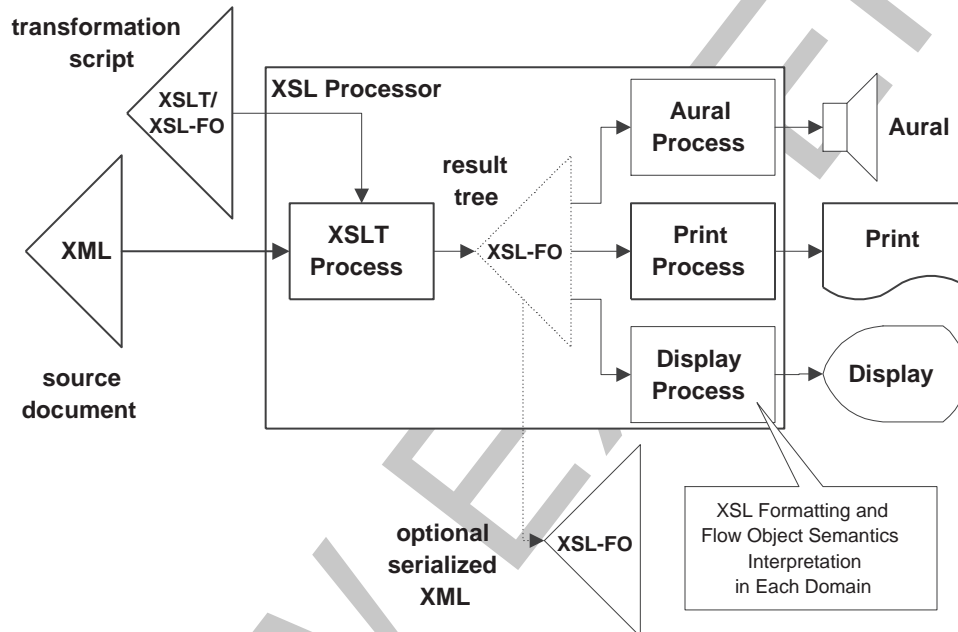
# Transforming and rendering XML information using XSLT and XSL-FO

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations

When the XSLT result tree is specified to utilize the XSL-FO formatting vocabulary:

- the normative behavior is to interpret the result tree according to the formatting semantics defined in XSL for the XSL-FO formatting vocabulary
- an inboard XSLT processor can effect the transformation to an XSL-FO result tree
- the XSL-FO result tree need not be serialized in XML markup to be conforming to the recommendation (though useful for diagnostics to evaluate results of transformation)



Of note:

- the stylesheet contains only the XSLT transformation vocabulary, the XSL formatting vocabulary, and extension transformation or foreign object vocabularies
- the source XML contains the user's vocabularies
- the result of transformation contains exclusively the XSL formatting vocabulary and any extension formatting vocabularies
  - does not contain any constructs of the source XML or XSLT vocabularies
- the rendering processes implement for each medium the common formatting semantics described by the XSL recommendation
  - for example, space specified before blocks of text can be rendered visually as a vertical gap between left-to-right line-oriented paragraphs or aurally as timed silence before vocalized content

# Using XSL-FO as an intermediate form

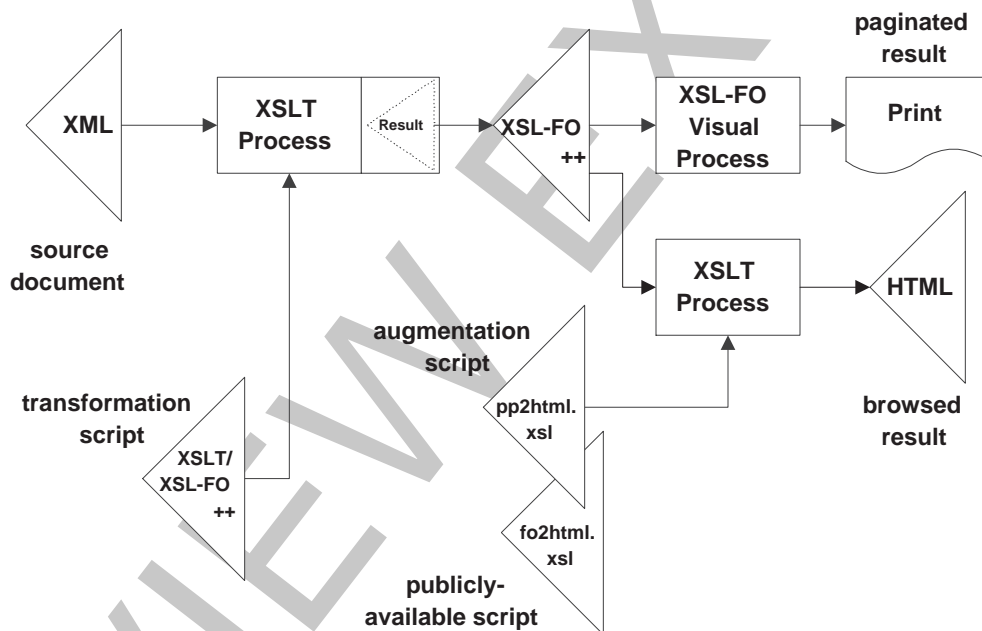
Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



XSL-FO can be used as the basis upon which to build an HTML result:

- the HTML page is an echo of the PDF page
- the HTML page is painted using only `<div>` and `<span>` and formatting properties to reproduce the appearance of the document
- no attempt to utilize HTML constructs such as `<h1>` or `<li>`
- one can introduce augmentations into the XSL-FO (referred to in the diagram as XSL-FO++) using a foreign namespace not recognized by an XSL-FO processor
- the augmentations are ignored by the standard processor
  - errors are not triggered
- the augmentations can be interpreted by a script that recognizes the foreign namespace
- the standard XSL-FO is translated to HTML using the publicly-available script where not overridden by the augmentation script
- the use of augmentations is entirely optional
  - the standard XSL-FO to HTML conversion handles much of XSL-FO as is



Of note:

- find the latest version of the `fo2html.xsl` stylesheet using a Google search with the following criteria:
  - `fo2html site:renderx.com`



# Generating XSL-FO instances

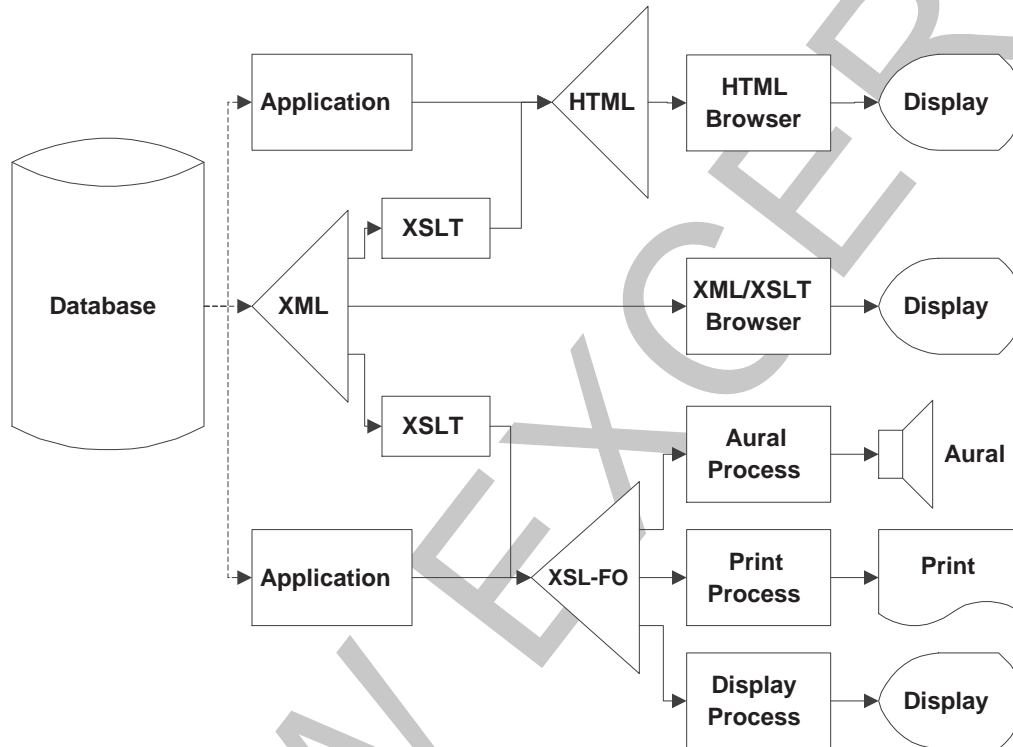
Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



The XSL vocabulary need not be created using XSLT

- volume HTML is often generated directly from applications
- XSLT can be used to transform XML into any vocabulary
- nothing in XSL-FO prevents it from being generated directly from applications



Sole requirement of instance is the use of the XSL vocabulary namespace:

- <http://www.w3.org/1999/XSL/Format>
- can be the default namespace
  - no XSL-FO attribute problems as exhibited with XSLT attributes when using the default namespace for XSLT

# Using XSL-FO on a server

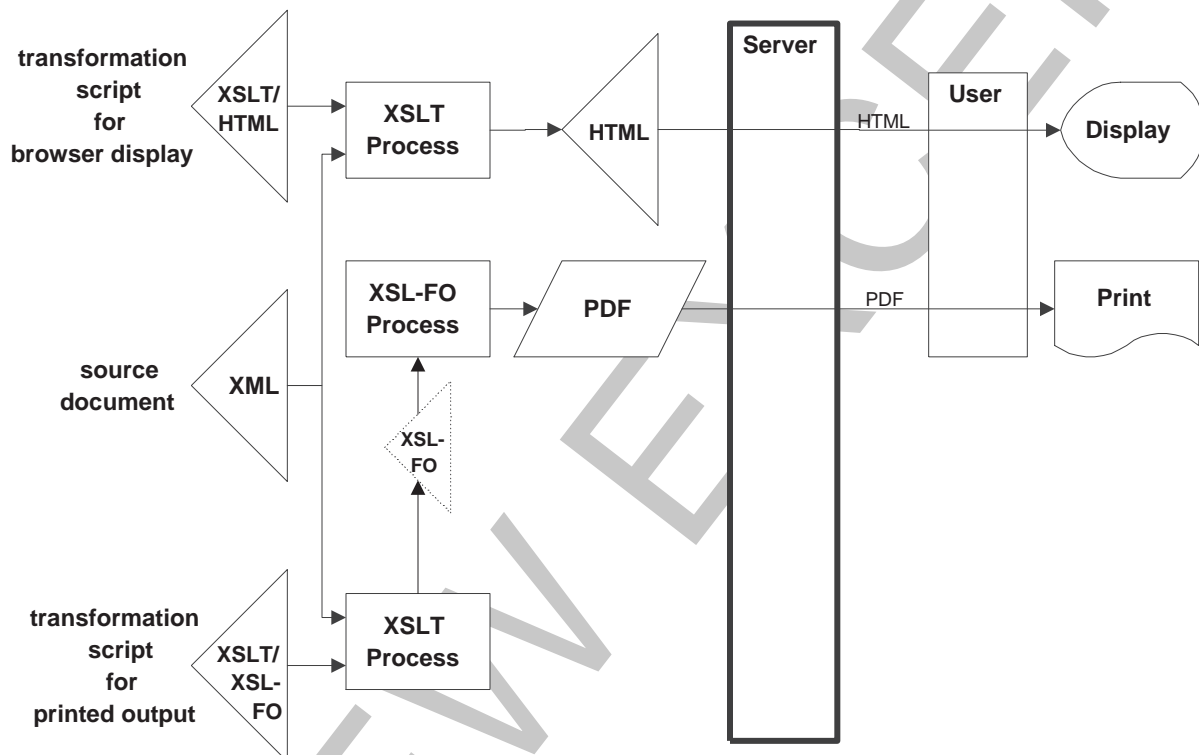
Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



A typical web-based use of XSL-FO is the server delivery of "printable versions" of information found on a web page

- a transformation from XML to HTML creates the static information for a browser
- a transformation from XML to XSL-FO to PDF creates the static information in printable form
  - user would use the PDF reader on their system to produce the paper through the system printer





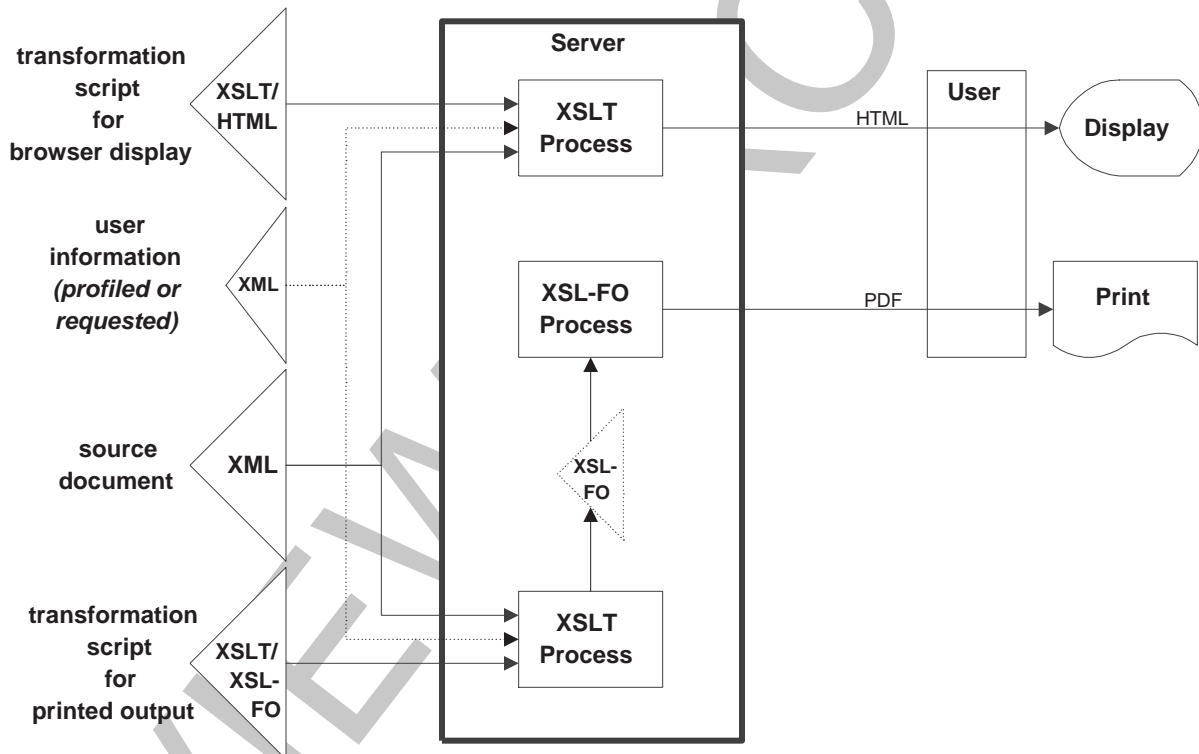
## Using XSL-FO on a server (cont.)

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations

A dynamic web-based use of XSL-FO is the on-the-fly synthesis and delivery of "printable versions" of information found on a web page

- static information is combined with user information based on the session engaged with the user
  - user information based on a predefined user profile
  - application information based on a dynamic request from the user
- a transformation from XML to HTML is used to deliver the information to a browser
- a transformation from XML to XSL-FO to PDF is used to deliver a paginated form of the information to the user
  - user would use the PDF reader on their system to produce the paper through the system printer



## Using XSL-FO on a server (cont.)

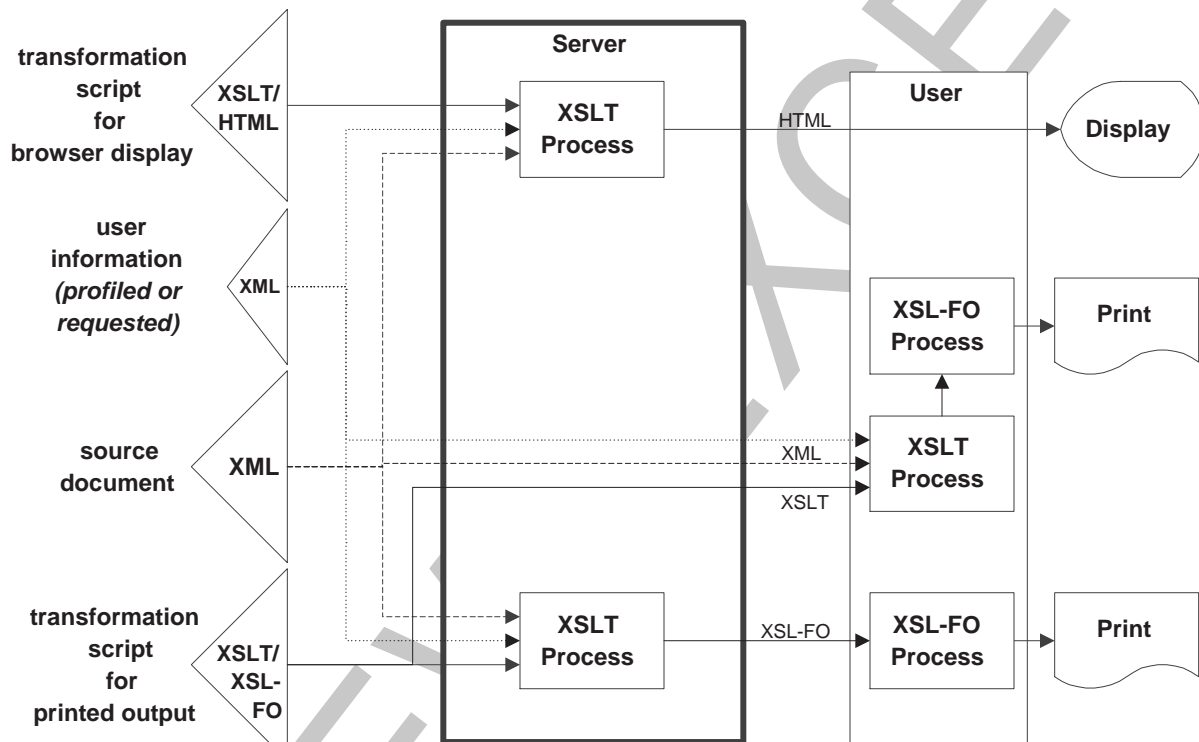
Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



A distributed web-based use of XSL-FO is to transform and produce on the target client or send XSL-FO "on the wire" to the target client software for processing into the resulting PDF

- similar benefits to dynamic web-based on-the-fly synthesis are enjoyed
- a transformation from XML to HTML is used to deliver the information to a browser
- XSL-FO-aware client software
- disadvantage in that the XSL-FO files are somewhat large if choosing to transmit them
- advantages are that the XSL-FO processing on the client can be fast and the server can use existing hardware assists for pure XSLT transformation





# Historical development of the XSL and XSLT Recommendations

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations

## Recommendation release history:

- first concept description floated in August 1997 with no official status within the World Wide Web Consortium (W3C)
  - <http://www.w3.org/TR/NOTE-XSL.html>
- the XSL Working Group officially chartered in early 1998
  - <http://www.w3.org/Style/XSL/>
- agreed upon requirements for XSL by the Working Group:
  - <http://www.w3.org/TR/WD-XSLReq>
- the XSL 1.0 Recommendation (XSL-FO) published October 15, 2001
  - <http://www.w3.org/TR/2001/REC-xsl-20011015/>
- the XSL 1.1 Recommendation (XSL-FO) published December 5, 2006
  - <http://www.w3.org/TR/2006/REC-xsl11-20061205/>
- the XSLT/XPath 1.0 Recommendations published November 16, 1999
  - <http://www.w3.org/TR/1999/REC-xslt-19991116>
    - <http://www.w3.org/1999/11/REC-xslt-19991116-errata> - errata
  - <http://www.w3.org/TR/1999/REC-xpath-19991116>
    - <http://www.w3.org/1999/11/REC-xpath-19991116-errata> - errata
- XSLT 1.1 (work abandoned)
  - <http://www.w3.org/TR/2000/WD-xslt11req-20000825> - requirements
  - <http://www.w3.org/TR/2001/WD-xslt11-20010824>
  - no incompatible changes to XSLT 1.0 in XSLT 1.1, only additional functionality
  - too many interactions with plans for XSLT 2.0, so functionality to be folded into XSLT 2.0 release
- XSLT 2.0/XPath 2.0/XQuery 1.0 published January 23, 2007
  - <http://www.w3.org/TR/2007/REC-xslt20-20070123/>
  - <http://www.w3.org/TR/2007/REC-xpath20-20070123/>
  - <http://www.w3.org/TR/2007/REC-xpath-datamodel-20070123/>
  - <http://www.w3.org/TR/2007/REC-xpath-functions-20070123/>
  - <http://www.w3.org/TR/2007/REC-xslt-xquery-serialization-20070123/>
  - <http://www.w3.org/TR/2007/REC-xquery-20070123/>
  - <http://www.w3.org/TR/2007/REC-xquery-semantics-20070123/>
  - <http://www.w3.org/TR/2007/REC-xqueryx-20070123>

## XSL information links

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



### Links to useful information

- <http://xml.coverpages.org/xsl.html> - Robin Cover
- <http://www.mulberrytech.com/xsl/xsl-list/> - mail list
- <http://www.dpawson.co.uk> - an XSL/XSLT FAQ
- <http://www.zvon.org/HTMLonly/XSLTutorial/Books/Book1/index.html> - numerous example XSLT scripts and fragments
- <http://www.openmath.org/cocoon/openmath/> - OpenMath project work by David Carlisle
- <http://www.CraneSoftwrights.com/links/trn-20080127.htm> - comprehensive XSLT/XPath and XSL-FO training material
- <http://www.CraneSoftwrights.com/resources> - free XSLT and XSL-FO resources
- <http://incrementaldevelopment.com/xsltrick/> - "Stupid XSLT Tricks"
- <http://xml.coverpages.org/xslSoftware.html> - list of tools
- <http://www.exslt.org/> - community effort for XSLT extensions
- <http://exslfo.sf.net> - community effort for XSL-FO extensions
- <http://foa.sourceforge.net/> - open source FO GUI authoring tool
- <http://www.xslfast.com/> - commercial FO GUI authoring tool
- <http://www.inventivedesigners.com/> - commercial FO GUI authoring tool
- <http://www.abisource.com/> - word processing with "Save As..." for XSL-FO
- <http://www.AntennaHouse.com/XSLsample/XSLsample.htm> - paginating XHTML
- ISBN 1-56609-159-4 - "The Non-Designer's Design Book", Robin Williams, Peachpit Press, Inc., 1994
- ISBN 0-8230-2121-1/0-8230-2122-X - "Graphic design for the electronic age; The manual for traditional and desktop publishing", Jan V. White, Xerox Press, 1988 (out of print but worthwhile to search for as a used book)



## XSL information links (cont.)

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations

### Examples of XSLT processors

- <http://www.jclark.com/xml/xt.html> - James Clark
- <http://saxon.sourceforge.net> - Mike Kay
- <http://msdn.microsoft.com/downloads/webtechnology/xml/msxml.asp> - updated web release of XML/XSLT processor for Internet Explorer 5 (IE6 follows the W3C specifications)
  - <http://www.netcrucible.com/xslt/msxml-faq.htm> - useful FAQ
- <http://www.altova.com/> - Altova
- <http://technet.oracle.com/tech/xml/> - Oracle
- <http://xml.apache.org/xalan/index.html> - Apache Project JAVA-based implementation (originally from IBM/Lotus AlphaWorks)
- <http://alphaworks.ibm.com/tech/LotusXSL> - IBM/Lotus AlphaWorks wrapper for Xalan
- <http://www.xmlsoft.org> - XSLT for Gnome
- <http://www.SolaceSystems.com> - XSLT-dedicated hardware (board/chip)
- <http://www.DataPower.com> - XSLT-dedicated hardware (box)
- <http://www.sarvega.com> - XSLT-dedicated hardware (box)
- <http://www.intel.com/software/xml> - Intel XML Appliance
- <http://www.ambrosoft.com/gregor.html> - XSLT compiler
- <http://www.infoteria.com> - iXSLT - commercial implementation
- <http://www.unicorn-enterprises.com/> - Unicorn XSLT Processor
- <http://www.a-dos.com> - XSLT processor and associated tools

The above list is just some of the early or interesting processors of the very many that are available commercially and publicly.

## XSL information links (cont.)

Chapter 2 - The context of XSL-FO

Section 1 - The XML family of Recommendations



### Examples of XSL formatting object rendering processors

- <http://www.AntennaHouse.com/> - AntennaHouse Windows-based and multi-platform versions
- <http://www.RenderX.com/> RenderX - direct to PDF
- <http://www.xmlpdf.com/ibex.html> - Ibex - direct to PDF
- <http://www.compart.net> - DOPE - direct to AFP or PDF
- <http://xml.apache.org/fop/> - FOP - direct to PDF, PCL, others
- <http://xmlroff.sourceforge.net/> - open source to PDF
- <http://www.Arbortext.com> - Epic and 3B2 composition tools
- <http://www.Adobe.com> - Adobe Document Server
- <http://www.alt-soft.com> - .Net - direct to PDF
- <http://www.Lunasil.com> - Java/COM - direct to PDF
- <http://www.tei-c.org.uk/Software/passivetex/> - Passive TeX - TeX to PDF
- <http://www.unicorn-enterprises.com/> - Unicorn UFO - TeX to PDF
- <http://www.alphaworks.ibm.com/tech/xfc> - IBM XFC - direct to PDF
- <http://www.xmlmind.com/foconverter> - Pixware XFC - XSL-FO to RTF
- <http://www.jfor.org/> - XSL-FO to RTF
- <http://www.xsmiles.org/> - XML browser using FOP

The above list is just some of the early processors of what is anticipated to be very many that will be available commercially and publicly.





# Hello world example

Chapter 2 - The context of XSL-FO  
Section 2 - Examples

Consider a simple, but complete, XSL-FO instance `hellofo.fo` for an A4 page report:

```

01 <?xml version="1.0" encoding="UTF-8"?>
02 <root xmlns="http://www.w3.org/1999/XSL/Format "
03     font-size="16pt">
04   <layout-master-set>
05     <simple-page-master
06       margin-right="15mm" margin-left="15mm"
07       margin-bottom="15mm" margin-top="15mm"
08       page-width="210mm" page-height="297mm"
09       master-name="bookpage">
10       <region-body region-name="bookpage-body"
11         margin-bottom="5mm" margin-top="5mm" />
12     </simple-page-master>
13   </layout-master-set>
14   <page-sequence master-reference="bookpage">
15     <title>Hello world example</title>
16     <flow flow-name="bookpage-body">
17       <block>Hello XSL-FO!</block>
18     </flow>
19   </page-sequence>
20 </root>

```

All examples illustrate instances of the XSL-FO vocabulary

- how the instance is created is not material to the semantics of the vocabulary
  - could be hand-authored in a simple text or XML editor
  - could be the result of an XSLT transformation from another XML vocabulary
  - could be output from any application
- the default namespace is used in the examples for brevity and clarity

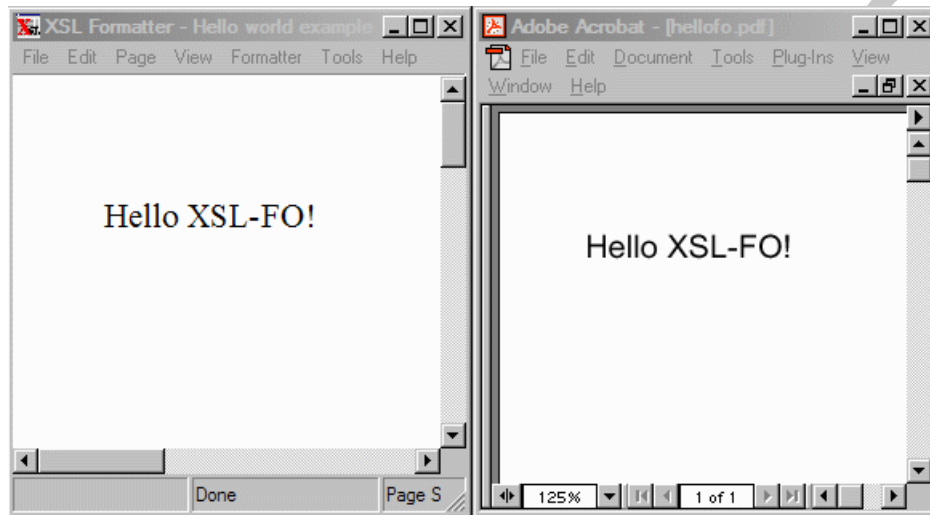
# Hello world example (cont.)

Chapter 2 - The context of XSL-FO  
Section 2 - Examples



Rendered on the screen in two adjacent windows using conforming XSL-FO processors:

- Antenna House XSL Formatter (an interactive XSL-FO rendering tool)
- Adobe Acrobat (a Portable Document Format (PDF) display tool)
- PDF created by RenderX XEP (a batch XSL-FO rendering tool)



Note the two renderings are not identical

- XSL-FO only specifies the formatting process, not the rendering process
- the XSL-FO instance may be insufficient in describing the entire formatting intent
- page fidelity is not guaranteed if the instance does not express the entire intent
  - XSLT semantics are extensive, but not necessarily comprehensive for certain nuances of formatting
- the rendering may engage certain property values of its own choosing

No different than two web browsers with different user settings for fonts

- a simple web page not using CSS properties relies on the browser settings
- the HTML constructs represent the intent of what is to be formatted
- absent formatting properties are satisfied using the tool option defaults

## Chapter 2 - The context of XSL-FO

### Section 2 - Examples

ains a mixture  
phs:

- excerpt of formatting objects created using an XSLT stylesheet and an XSLT stylesheet processor

+ISSN 1-894040-4-5; e-mail: [PERJY@XCF.MIACENT](mailto:PERJY@XCF.MIACENT) Introduction to XSL-FO 3004.00.12 24:56WTF/AR

# Training material example

Chapter 2 - The context of XSL-FO  
Section 2 - Examples



This page's material as an instructor-led handout:

- excerpt of formatting objects created using an XSLT stylesheet and an XSLT stylesheet processor

```

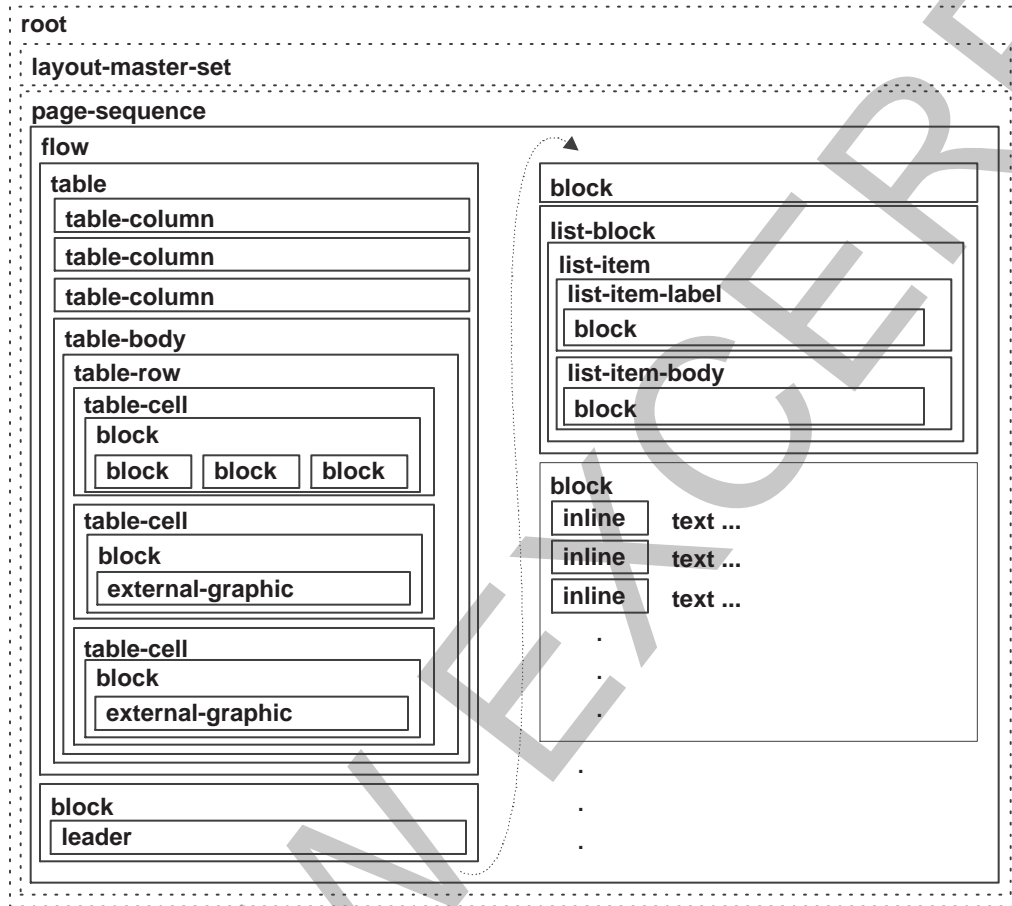
01 <flow flow-name="pages-body"><table>
02   <table-column column-width="( 210mm - 2 * 15mm ) - 2in"/>
03   <table-column column-width="1in"/>
04   <table-column column-width="1in"/>
05   <table-body><table-row><table-cell><block text-align="start">
06     <block font-size="19pt">Training material example</block>
07     <block font-size="10pt" space-before="10pt">Module
08 2 - The context of XSL-FO</block>
09     <block font-size="10pt">Lesson 2 - Examples</block></block>
10   </table-cell>
11   <table-cell><block text-align="end"><external-graphic
12     src="url(&quot;..\whitesml.bmp&quot;)" /></block></table-cell>
13   <table-cell><block text-align="start"><external-graphic
14     src="url(&quot;..\cranesml.bmp&quot;)" /></block></table-cell>
15 </table-row></table-body></table>
16 <block line-height="3px"><leader leader-pattern="rule"
17   leader-length="100%" rule-thickness="1pt"/></block>
18 <block space-before="6pt" font-size="14pt">
19 This page's material as an instructor-led handout:</block>
20 <list-block provisional-distance-between-starts=".43in"
21   provisional-label-separation=".1in" space-before="6pt">
22   <list-item relative-align="baseline">
23     <list-item-label text-align="end" end-indent="label-end()">
24       <block></block></list-item-label>
25     <list-item-body start-indent="body-start()">
26       <block font-size="14pt">excerpt of formatting objects created
27 using an XSLT stylesheet and an XSLT stylesheet processor</block>
28     </list-item-body></list-item></list-block>
29 <block space-before="12pt div 2" font-family="Courier"
30   linefeed-treatment="preserve" white-space-collapse="false"
31   white-space-treatment="preserve" font-size="12pt"><inline
32 font-size="inherited-property-value(font-size) div 2">01 </inline
33 >&lt;flow flow-name="pages-body"&gt;&lt;table&gt;
34 <inline font-size="inherited-property-value(font-size) div 2"
35 >02 </inline> &lt;table-column column-width...
```



# Training material example (cont.)

Chapter 2 - The context of XSL-FO  
Section 2 - Examples

The nesting of the hierarchy of the formatting objects in the example page:



## Chapter 3 - Basic concepts of XSL-FO



- 
- Introduction - Essential concepts and terminology
  - Section 1 - Basic concepts
  - Section 2 - Processing model
  - Section 3 - Formatting object XML vocabulary
  - Section 4 - Groupings of formatting objects for flow

PREVIEW EXCERPT

# Essential concepts and terminology

Chapter 3 - Basic concepts of XSL-FO



## Layout-based vs. content-based formatting

- layout-based formatting respects the constraints of the target medium
  - limitations or capacities of target may constrain content or appearance
- content-based formatting respects the quantity and identity of the information
  - generates as much of the target medium as required to accommodate the information

## Formatting is different than rendering

- expressing what you want done vs. expressing how it is accomplished
- similar to difference between declarative and imperative programming
- similar to difference between XSLT "transformation by example" and other approaches of "transformation by algorithm"

## Differing processing model concepts are expressed using unambiguous terminology

- an XSL-FO instance with elements and attributes
- a formatting object tree with objects and properties
- a refined formatting object tree with objects and area traits
- an area tree with areas and traits

## Formatting model and vocabulary properties extend what is currently available for web presentation

- adds arbitrary boundaries as pages instead of infinite-length canvas in a browser
- inspired by DSSSL but diverges more closely to CSS2
- classes of XSL-FO properties:
  - CSS properties by copy (unchanged CSS2 semantics)
  - CSS properties with extended values
  - CSS properties "broken apart" to a finer granularity
  - XSL-FO-specific properties
- supports multiple writing directions and reference orientations

## Essential concepts and terminology (cont.)

Chapter 3 - Basic concepts of XSL-FO



---

The XSL-FO objects covered in this chapter are:

- `<root>` (6.4.2)
  - the document element of the XSL-FO instance
- `<layout-master-set>` (6.4.7)
  - the collection of definitions of page geometries and page sequencing and selection patterns
- `<page-sequence>` (6.4.5)
  - the specification of that information to be paginated over a sequence of pages with common static information
- ① `<page-sequence-wrapper>` (6.4.6)
  - the specification of inherited properties across a number of page sequences
- `<flow>` (6.4.19)
  - the content that is flowed to as many pages as required and formatted according to the appearance properties





# Layout-based vs. content-based formatting

Chapter 3 - Basic concepts of XSL-FO  
Section 1 - Basic concepts

Layout-based formatting accommodates the medium

- often uses absolute positioning, sizing and page counting
  - a magazine may need a particular columnist's article to appear on the right-hand edge of page 7 and the three lead stories within the first four pages
- rules of layout dictate the quantity and appearance of content
- typically unstructured authoring and formatting
  - desktop publishing, journalism, etc.

Content-based formatting accommodates the information

- rules of content dictate the layout of information (rule-based formatting)
- 40,000 to 60,000 pages in a single aircraft maintenance manual cannot be individually formatted
- typically highly-structured authoring and formatting
  - technical publications: pharmaceutical, aerospace, automotive, etc.

XSL-FO oriented more to content-based formatting than layout-based formatting

- because XSL-FO is flow-based not page-based
  - page breaks are triggered by the need of content flowed into a page sequence
  - the stylesheet writer does not lay out every page in an XSL-FO stream
- a "turn the crank" approach, not a "tweak and peek" approach
  - mechanical accommodation of content
- limited support of the order of specific page sequences
  - high-caliber copy-fitting often cannot be mechanical or unattended

XSL-FO is not oriented to loose-leaf publishing

- no inherent maintenance facilities for past versions of individual pages
- no inherent support of change pages (a.k.a. "A pages")
- no inherent support of lists of effective pages

# Formatting vs. Rendering

Chapter 3 - Basic concepts of XSL-FO  
Section 1 - Basic concepts



An XSL-FO instance describes the intent of how a stream of information is to be formatted

- XSL-FO instance typically generated by a stylesheet acting on an instance of XML information
  - the original vocabulary of the XML source is transformed into the XSL-FO vocabulary
    - the #PCDATA and attribute content is rewrapped in formatting vocabulary
  - resulting transformed file is useful for diagnostics when reified as syntax
  - opportunity for store and forward applications of formatting intent
- no feedback loop from the formatter to the stylesheet (unlike interactive agents)
  - XSL-FO information must be complete with respect to all desired behaviors of the formatter
  - special formatting cases or conditions can be accommodated through contingencies expressed in the XSL-FO semantics

Information to be formatted is repackaged from the source vocabulary into the formatting vocabulary

- information arranged in a collection of formatting objects, each specifying a part of:
  - layout
    - the location of the information positioned on the target medium
    - areas defined by the user and located within other areas
      - hierarchical tree of rectangles on each page
  - appearance/impartation
    - the conveyance of the information to the user
    - visual formatting (font, size, color, weight, etc.)
    - aural synthesis (voice, volume, azimuth, pitch, etc.)
  - pagination/flow
    - the parceling of a stream of content within the layout areas
    - generating areas as required to accommodate the quantity of information
- objects are expressed in an XSL-FO XML instance as elements

A complete alphabetical list of formatting objects is in Annex C XSL-FO object summary (page 140).

- it is not necessary to know all objects to get effective results

## Formatting vs. Rendering (cont.)

Chapter 3 - Basic concepts of XSL-FO  
Section 1 - Basic concepts



---

Formatter responsible for interpreting intent to be rendered

- following the Recommendation the formatter determines what is to be rendered
  - how the formatter interprets the formatting objects is defined in excruciating detail
- some properties are specifically targeted to certain media

The semantics of rendering are not described in detail

- device-specific rendition based on semantics of formatting objects
- how the rendering agent accomplishes the task of effecting the result of formatting is entirely up to the agent
  - must produce the same result as the intent described by the Recommendation

Rendering, itself, may be a multiple-step process

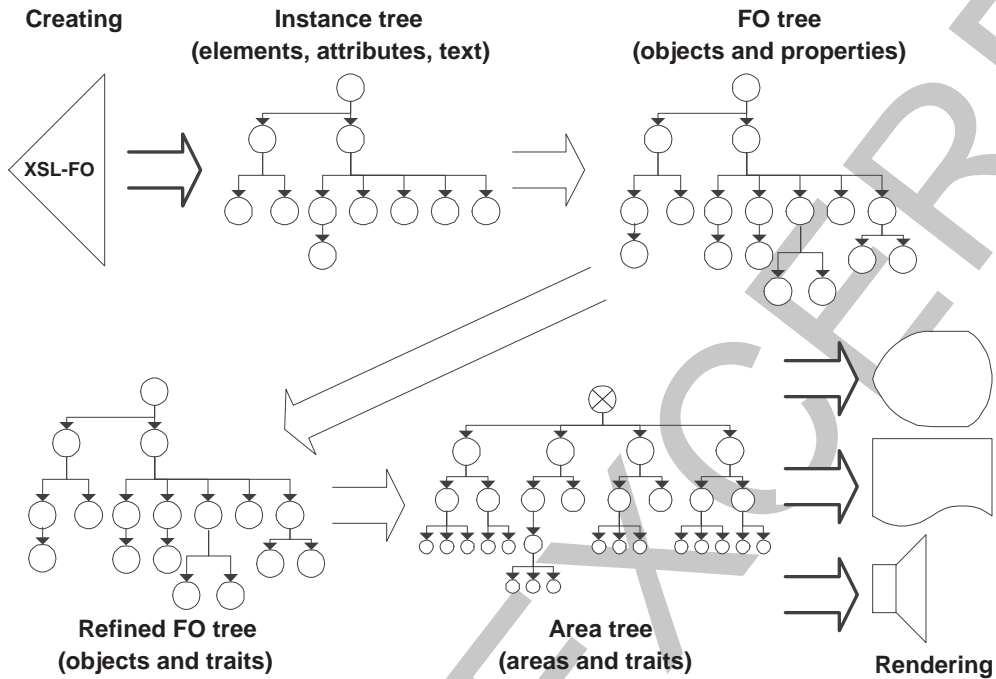
- production of another intermediate formatting language
  - e.g. TeX
- production of an electronic representation of the final form page description
  - e.g. Portable Document Format (PDF) by Adobe
  - e.g. Standard Page Description Language (SPDL) - ISO/IEC 10180
- interpretation and rendering to the physical final form
  - e.g. paper from the page description language
- could be many steps to final result
  - e.g. XML to XSL-FO to TeX to PDF to paper

# Processing model of formatting

Chapter 3 - Basic concepts of XSL-FO  
Section 2 - Processing model



From formatting objects to area tree in a well-defined process:





## Processing model of formatting (cont.)

Chapter 3 - Basic concepts of XSL-FO  
Section 2 - Processing model

Formatting vocabulary represents typographical abstractions available to the designer

- documented as semantics in the XSL-FO Recommendation
- expressed as the intent of the designer for the formatting of the page
  - in elements, attributes and text using the XSL-FO vocabulary
- may be created outside of the formatter in an XML instance of XSL-FO
- information represented within the formatter as an Instance Tree

Formatting objects represent the specifications of layout, appearance and pagination

- information interpreted as the Formatting Object Tree
- most elements and text become formatting objects
  - in-stream foreign object elements do not become objects
  - certain white-space-only text nodes do not become objects
  - constructs not from the XSL-FO namespace that are used in <declarations> do not become objects
- attributes become properties
  - any property can be specified on any element

Properties are refined to traits where applicable for each formatting object

- information interpreted as the Refined Formatting Object Tree
- a single property may become one or more traits of any kind
  - formatting traits (e.g. size and position)
  - rendering traits (e.g. style and appearance)
- shorthand expansion into individual granules of properties
- expression evaluation for computed properties
  - example: property of "2em" becomes a trait of "40pt" if the font size is "20pt"
- some traits are inherited from ancestral property specifications
  - some properties are not inherited but can be explicitly inherited on request
  - inapplicable properties are removed from the refined tree
- once all traits that are applicable to formatting objects are determined, all traits not applicable to a given object are removed

## Processing model of formatting (cont.)

Chapter 3 - Basic concepts of XSL-FO  
Section 2 - Processing model



Geometric areas on pages are returned from interpretation of formatting objects and their traits

- information interpreted as areas in the Area Tree
  - the formal theoretical expression of the semantics represented by the vocabulary
- the interpretation of some objects does not return any areas
- the interpretation of some objects returns exactly one area
- the interpretation of some objects returns as many areas as needed by descendent objects
- all areas in XSL-FO 1.0 are rectangular or square
- each area has geometric page position, z-layer position, content, background, padding, borders
- a stylesheet writer has the desired layout of areas in mind and determines which objects will return the desired areas in the desired order
- page areas are ordered but not geometrically related

Rendering agent effects the impartation of areas according to the medium

- Recommendation gives guidelines for each area and trait for visual and aural media
  - some traits are only applicable to interactive media
- some missing trait values can be arbitrarily inferred by the rendering agent (e.g. font, volume)
- the process of rendering is not specified in the Recommendation

Recommendation is written to lead a formatter implementer in carrying out requirements

- some traits are boolean values targeted to the implementer and reflecting an area's role or relative order to other areas
- rigor of Recommendation needed for finely-tuned typographical nuances
  - not written to lead the stylesheet writer in writing a stylesheet
  - many very complex interactions are not important to many stylesheets
- simple things can be done simply



## Vocabulary structure

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary

The semantics of a formatting object are defined in terms of areas and their traits

- which areas the object generates (if any)
- where the areas are placed in the area tree hierarchy
- interactions with areas from and specifications for other objects
- see Objects summarized by name (page 142) for a list of all objects
- see Prototypical hierarchy (page 147) for a prototypical hierarchy

Formatting objects are specified using elements, attributes, and text

- may be a stand-alone XML instance being interpreted by a formatter
- may be a node tree passed as the result of transformation

XSL-FO elements must use the URI "`http://www.w3.org/1999/XSL/Format`" for the namespace

- extension elements *must not* use the XSL-FO URI
- extension attributes *must* have a non-null prefix
- extension constructs must be ignored if not recognized by an XSL-FO processor
- the default namespace may be used for the XSL-FO elements
- un-prefixed attributes are assumed to be in the vocabulary of the element to which they are attached

Element-type and attribute names:

- all letters are lowercase
- a hyphen separates multiple words in a single name
- a dot separates names from their compound components
- some abbreviations are used if the name is already used in XML or HTML

Examples:

- `space-before="12pt"`
  - all compound components are set to their initial values
- `space-before.conditionality="retain"`
  - sets the individual compound component to retain the space if at the beginning or termination of a reference area



## Vocabulary structure (cont.)

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



Every property is allowed to be specified on every object

- provides for inheritance in the formatting object tree during refinement
- only those traits meaningful to a formatting object are in the refined formatting object tree
  - utilized by some or all of the areas created by the object
- see Property summary (page 180) for a list of all properties

Every object may have a unique identifier specified using `id=` property

- used as the target of a reference from other objects
- value is not inherited from other objects
- value is assigned to the first child area generated for the object
  - if the object doesn't generate any areas, the identifier is lost

Extension objects and/or properties are allowed in any other namespace

- e.g. `xmlns:axf="http://www.antennahouse.com/names/XSL/Extensions"`
  - identifies extension functionality described by Antenna House, Inc.
- e.g. `xmlns:rx="http://www.renderx.com/XSL/Extensions"`
  - identifies extension functionality described by RenderX
- e.g. `xmlns:ibex="http://www.xmlpdf.com/2003/ibex/Format"`
  - identifies extension functionality described by Ibex
- e.g. `xmlns:dope="http://www.compart.net/xmlns/cpfo"`
  - identifies extension functionality described by Compart DOPE
- e.g. `xmlns:at="http://www.arbortext.com/namespace/XslFoExtensions"`
  - identifies extension functionality described by Arbortext
- no restrictions on who or which tools can support extended functionality
- important portability issue
  - no obligation for any tool to support any extension functionality

Non-XSL-FO namespaces are allowed for embedded graphic images

- e.g. `xmlns:svg="http://www.w3.org/2000/svg"`
  - the W3C SVG 2-dimensional graphics vocabulary
- e.g. `xmlns:math="http://www.w3.org/1998/Math/MathML"`
  - the W3C Mathematical markup language





# Direct vs. constraint property specification

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary

Properties influence the behavior of an object in the creation of areas for the formatted result

- properties are specified in an XSL-FO XML instance or node tree as attributes
  - properties become traits in the area tree
- strategic use can promote maintainability or consistency
  - placement in the hierarchy for inheritance
  - specified algorithmically for contextually sensitive behavior
- some properties directly specify a formatting result
  - e.g. `color=` property
  - the foreground color is specified regardless of the context in the area tree
- some properties constrain the formatting result based on an interpretation of context in the area tree
  - e.g. `space-before=` property
  - the amount of space before the area is constrained to the value specified but may be discarded if the area's conditionality allows it to be discarded and the area is at the beginning of a reference area
- compound properties are comprised of components for fine-grained specification
  - often originate from the CSS2 specification but specify too many aspects of formatting
  - e.g. `space-before.optimum=`, `space-before.minimum=`, `space-before.maximum=`, `space-before.conditionality=` and `space-before.precedence=` components
- properties can influence either layout or appearance or both

A complete alphabetical list of formatting properties is in Annex D XSL-FO property summaries (page 156).

- it is not necessary to know all properties to get effective results

# Property value expressions

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



A property's value can be the evaluation of an expression

- may include fixed values
  - e.g. `space-before="24pt div 2"`
- may include contextually-sensitive values
  - e.g. `space-before="from-parent(font-size) div 2"`
- same operators as in XPath 1.0
- includes same operands as in XPath 1.0
- includes length values as operands that are not allowed in XPath 1.0
- expressions influenced by the font size evaluate the font size before evaluating any other components of the expression

Core function library defined for property expressions

- functions with access to property values of the current node
  - e.g. `inherited-property-value(property-name)`
    - this obtains a value that may be specified on the current node or may be inherited from the closest ancestral node that specifies the value
- functions with access to property values of other nodes
  - e.g. `from-parent(property-name)`
    - this looks for the property value directly from the parent object specification
  - e.g. `merge-property-values(property-name)`
    - this looks for the property value directly from the particular sibling object specification corresponding to a given state of the user interface
- summarized in Functions summarized by name (page 137)

Numerous property data types

- summarized in Property data types (page 168)
- simple-valued and compound-valued values

Numerous functions available for expressions

- summarized in Functions summarized by name (page 137)
- numeric functions can be applied to length values by reducing the "unit power" and adding it back again after
  - a length has a unit power of 1, while a number has a unit value of zero
  - e.g. `round()` takes a number argument and not a length argument
  - can use: `round( length-value div 1.0cm ) * 1.0cm`



# Inherited properties

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary

Inheritable properties need not be specified for a formatting object

- when not specified, the ancestry of the formatting object tree is examined for the closest specification
- all inheritance completed during refinement before the area tree is created
- see Inherited properties (page 176) for a summary

Inheritance goes "through" block-level or out-of-line constructs

- recall the nature of the processing model (see page 50) where refinement happens on the object tree
- often the source of unexpected results in a stylesheet's result
- e.g. the same `start-indent` property used to indent a table is inherited by the table's cell's content
- e.g. a footnote's body inherits properties from the block in which the footnote is specified

Some functions can obtain property values from outside the ancestry

- the attribute can specify the result of a function call instead of a fixed value
- e.g. `merge-property-values()`
- e.g. `from-table-column()`

Some properties look like they are inherited, but do not go "through" out-of-line constructs

- e.g. `text-decoration`
  - when specified on a block-level construct, it affects all inline descendants of the block, but not any descendants of any out of line areas found in the block such as float or footnote bodies (though footnote inline areas are affected)

# Shorthand properties

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



Shorthand properties specify a number of standalone properties using a single specification

- e.g. `font="italic small-caps bold 40pt/50pt Courier"`
  - one specification sets each of the following properties in order
    - `font-style=`
    - `font-variant=`
    - `font-weight=`
    - `font-size=`
    - `line-height=`
    - `font-family=`
- different than a compound property
  - a shorthand property represents a set of standalone properties
  - a compound property is comprised of a set of components
  - one cannot specify a compound property on a shorthand property
- individual properties of a shorthand property are supported severally
  - shorthand properties can be used in conjunction with individual properties
  - those that are specified are more precise than those that are inferred
- shorthand properties are processed in increasing precision
  - precedence order independent of attribute specification order
  - more-specific attribute name has higher precedence than less-specific attribute name
    - `border-color=` is more precise than `border=`
    - e.g. `border-bottom-color=` is more precise than `border-color=`
  - border named-edge specifications have higher precedence than generic border specifications
    - e.g. `border-bottom=` is more precise than `border-style=`

A processor is not obliged to support shorthand properties

- see Shorthand properties (page 179) for a summary
- portability problem when developing a stylesheet using one processor and delivering the stylesheet to someone else using a different processor



# Conformance

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary

---

Three levels of conformance that can be claimed by a processor:

- complete
  - entire Recommendation is supported
- extended
  - no support for shorthand properties
  - very few esoteric positioning- and font-related properties not supported
  - remainder of Recommendation fully supported
- basic
  - minimum level of pagination or aural rendering support
  - fallback behaviors defined for unsupported extended facilities
- implementing fallback behaviors cannot be construed as being complete conformance

In a given medium, a processor must support all formatting objects and properties specified for the conformance level claimed

- conformance levels are described for two rendering class distinctions
  - visual media
  - aural media
- can implement fallback processing in other media
  - conformance qualification does not include fallback processing

# Top-level formatting objects

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



All XSL-FO structures must have the following formatting objects:

- `<root>`
  - the document element of an XSL-FO structure
  - handy location for document-wide inherited attributes
- `<layout-master-set>`
  - the collection of page geometries available to use for pagination
  - also collects definitions of patterns of geometry sequencing for nuances in changes of page layout

All XSL-FO structures must have at least one page sequence with flowed content to be paginated

- `<page-sequence>`
  - begin a new page with some content to be paginated
  - defines properties and content common to all pages in the sequence
    - e.g. the formatting of page numbers in the page sequence
    - e.g. the definitions of static content (i.e. headers, footers, etc.) for the page geometries selected for the page sequence
  - as many pages will be created as is needed to accommodate the flow given for the sequence
    - an exception is when the number of pages of the sequence or the starting page number of the following sequence is constrained to be an even or odd number
      - the formatter will generate a blank page to accommodate the criterion
      - the blank page can be defined with static content to inform the reader the page is intentionally left blank
    - the use of `break-before=` and `break-after=` in a sequence can cause information to start at the top of a page in the middle of the sequence
  - the formatter will use either the page geometry or the pattern of geometry sequencing to select the pages, in order, to accommodate the content and any blank pages
- ① `<page-sequence-wrapper>`
  - introduce a branch in the formatting object tree for inherited properties applicable to descendant objects
  - this formatting object does not generate any output itself and is used only as a convenience for property specification
- `<flow>`
  - the content to be paginated in a sequence of pages of a given geometry or geometry sequence
  - can only contain block-level child formatting objects as its top-level objects

## <root> Object

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



### Purpose:

- the document element of an XSL-FO instance
- returns areas generated by children as a sequence of page-viewport-areas
  - such areas are ordered but there is no geometric relationship between areas

### Content:

- ¶ (6.4.2) (layout-master-set,declarations?,page-sequence+)
- ¶ (6.4.2) (layout-master-set,declarations?,bookmark-tree?,(page-sequence|page-sequence-wrapper)+)
- Child objects (alphabetical):
  - ¶ <bookmark-tree>(6.11.1;113)
  - <declarations>(6.4.3;119)
  - <layout-master-set>(6.4.7;63)
  - <page-sequence>(6.4.5;65)
  - ¶ <page-sequence-wrapper>(6.4.6;67)

### Property sets:

- Common Accessibility Properties(7.5;159)

### Other optional properties:

id=(7.30.8;209)

¶ index-class=(7.24.1;209)

¶ index-key=(7.24.2;209)

media-usage=(7.27.11;216)

### Property of interest:

- media-usage= used when establishing either pagination with bounded page length or no pagination with infinite page length

## <root> Object (cont.)

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary




---

The XSL-FO instance document element:

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <root xmlns="http://www.w3.org/1999/XSL/Format" font-size="16pt">
03   <layout-master-set>
04     <simple-page-master master-name="bookpage"
05       page-height="297mm" page-width="210mm"
06       margin-top="15mm" margin-bottom="15mm"
07       margin-left="15mm" margin-right="15mm">
08       <region-body region-name="bookpage-body"
09         margin-top="5mm" margin-bottom="5mm"/>
10     </simple-page-master>
11   </layout-master-set>
12   <page-sequence master-reference="bookpage">
13     <title>Hello world example</title>
14     <flow flow-name="bookpage-body">
15       <block>Hello XSL-FO!</block>
16     </flow>
17   </page-sequence>
18 </root>

```



## <layout-master-set> Object

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



### Purpose:

- the collection of definitions of page geometries, available regions, and page selection patterns

### Content:

- ¶ (6.4.6) (simple-page-master|page-sequence-master)+
- ¶ (6.4.7) (simple-page-master|page-sequence-master|flow-map)+
- Child objects (alphabetical):
  - ¶ <flow-map>(6.4.22;105)
  - <page-sequence-master>(6.4.8;105)
  - <simple-page-master>(6.4.13;81)
- Referring object:
  - <root>(6.4.2;61)

No properties are defined for this formatting object.

## <layout-master-set> Object (cont.)

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



The set of layout masters:

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <root xmlns="http://www.w3.org/1999/XSL/Format" font-size="16pt">
03   <layout-master-set>
04     <simple-page-master master-name="bookpage"
05       page-height="297mm" page-width="210mm"
06       margin-top="15mm" margin-bottom="15mm"
07       margin-left="15mm" margin-right="15mm">
08       <region-body region-name="bookpage-body"
09         margin-top="5mm" margin-bottom="5mm"/>
10     </simple-page-master>
11   </layout-master-set>
12   <page-sequence master-reference="bookpage">
13     <title>Hello world example</title>
14     <flow flow-name="bookpage-body">
15       <block>Hello XSL-FO!</block>
16     </flow>
17   </page-sequence>
18 </root>

```

## <page-sequence> Object

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



### Purpose:

- the definition of information for a sequence of pages with common static information
- generates as many pages as will fit the flowed content that is supplied
- each new page sequence begins the flow at the top of a new page
  - e.g. starts of chapters, front matter, back matter, etc.

### Content:

- ¶ (6.4.5) (title?,static-content\*,flow)
- ¶ (6.4.5) (title?,folio-prefix?,folio-suffix?,static-content\*,flow+)
- Child objects (alphabetical):
  - <flow>(6.4.19;68)
  - ¶ <folio-prefix>(6.6.13;104)
  - ¶ <folio-suffix>(6.6.14;104)
  - <static-content>(6.4.20;104)
  - <title>(6.4.21;81)
- Referring objects:
  - <root>(6.4.2;61)
  - ¶ <page-sequence-wrapper>(6.4.6;67)

### Required property:

master-reference=(7.27.9;216)

### Optional properties:

country=(7.10.1;200)

¶ flow-map-reference=(7.27.19;204)

force-page-count=(7.27.6;207)

format=(7.26.1;207)

grouping-separator=(7.26.2;207)

grouping-size=(7.26.3;208)

id=(7.30.8;209)

¶ index-class=(7.24.1;209)

¶ index-key=(7.24.2;209)

initial-page-number=(7.27.7;210)

language=(7.10.2;212)

letter-value=(7.26.4;213)

¶ reference-orientation=(7.21.3;225)

¶ writing-mode=(7.29.7;238)

### Shorthand influencing the above properties:

¶ xml:lang=(7.31.24;239)

### Properties of interest:

- master-reference= selects the page geometry or sequencing of page geometry and the available regions within those page definitions
  - cannot change the page geometry based on content found in the flow
- format= governs how the page number is represented as a sequence of characters
  - numbers, roman numerals, etc.
  - formally defined by reference to XSLT(see page 175)

## <page-sequence> Object (cont.)

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



A sequence of pages:

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <root xmlns="http://www.w3.org/1999/XSL/Format" font-size="16pt">
03   <layout-master-set>
04     <simple-page-master master-name="bookpage"
05       page-height="297mm" page-width="210mm"
06       margin-top="15mm" margin-bottom="15mm"
07       margin-left="15mm" margin-right="15mm">
08       <region-body region-name="bookpage-body"
09         margin-top="5mm" margin-bottom="5mm"/>
10     </simple-page-master>
11   </layout-master-set>
12   <page-sequence master-reference="bookpage">
13     <title>Hello world example</title>
14     <flow flow-name="bookpage-body">
15       <block>Hello XSL-FO!</block>
16     </flow>
17   </page-sequence>
18 </root>

```

## <page-sequence-wrapper> Object

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



### Purpose:

- a formatting object used only as a home for inherited properties applicable to descendent formatting objects
- useful when formatting properties apply to some page sequences but not all

### Content:

- ① (6.4.6) (page-sequence|page-sequence-wrapper)\*
- Child objects (alphabetical):
  - <page-sequence>(6.4.5;65)
  - ① <page-sequence-wrapper>(6.4.6;67)
- Referring objects:
  - <root>(6.4.2;61)
  - ① <page-sequence-wrapper>(6.4.6;67)

### Optional properties:

id=(7.30.8;209)

① index-key=(7.24.2;209)

① index-class=(7.24.1;209)

Adds a branch in the formatting object tree (as shown on page 50)

- opportunity to add inherited properties for all pages in all page sequences

## <flow> Object

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



### Purpose:

- the content that is flowed to as many pages as required to fit
  - flowed content is not repeated by the formatter
- the length of the flow governs the length of the page sequence

### Content:

- (6.4.19) (%block;)+
- Child object:
  - %block;(6.2;71)
- Referring object:
  - <page-sequence>(6.4.5;65)
- may begin with any number of <marker> children

### Required property:

flow-name=(7.27.5;204)

### Optional properties:

id=(7.30.8;209)

① index-key=(7.24.2;209)

① index-class=(7.24.1;209)

### Property use:

- flow-name= indicates into which area on the page the child information is to be flowed
  - area does not have to exist on the particular page geometry being rendered
    - the method to ignore a flow on particular pages

## <flow> Object (cont.)

Chapter 3 - Basic concepts of XSL-FO

Section 3 - Formatting object XML vocabulary



A flow of areas for a region:

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <root xmlns="http://www.w3.org/1999/XSL/Format" font-size="16pt">
03   <layout-master-set>
04     <simple-page-master master-name="bookpage"
05       page-height="297mm" page-width="210mm"
06       margin-top="15mm" margin-bottom="15mm"
07       margin-left="15mm" margin-right="15mm">
08       <region-body region-name="bookpage-body"
09         margin-top="5mm" margin-bottom="5mm"/>
10     </simple-page-master>
11   </layout-master-set>
12   <page-sequence master-reference="bookpage">
13     <title>Hello world example</title>
14     <flow flow-name="bookpage-body">
15       <block>Hello XSL-FO!</block>
16     </flow>
17   </page-sequence>
18 </root>

```

# Groupings of formatting objects for flow

Chapter 3 - Basic concepts of XSL-FO

Section 4 - Groupings of formatting objects for flow



## Stacking

- areas added to branches of the area tree interact with adjacent areas found in the tree
- areas stack relative to adjacent areas on the rendered result in different directions:
  - block-progression direction
    - e.g. top-to-bottom for the Western European writing direction
  - inline-progression direction
    - e.g. left-to-right for the Western European writing direction

## Block-level object

- stacks next to siblings in the block-progression direction
  - new blocks restart in the block-progression direction
- breaks the flow of information in the inline-progression direction
  - siblings cannot be "beside" each other in the inline-progression direction
  - e.g.: blocks, tables, lists, etc.
- the `<block>` object is a frequently used construct
  - e.g. for paragraphs, headings, captions, table cell contents, etc.

## Inline-level object

- stacks next to siblings in the inline-progression direction
- does not restart the flow of information in the block-progression direction
- specifies portions of content to be flowed into lines of the parent block where the portions are distinct from their sibling portions

## Neutral object

- an object that is allowed anywhere without impacting the stacking of siblings or progression direction of the parent

## Out-of-line object

- areas stack out of line to the areas of siblings of the object, without impacting the siblings of the object, by adding areas to different branches of the area tree than the siblings
- does not break the flow of information in any progression direction

## Out-of-line inline-level object

- a portion stacks next to the siblings of the object in the line-progression direction
- a portion stacks out-of-line to the siblings of the object, without impacting the siblings of the object





## Block-level objects

Chapter 3 - Basic concepts of XSL-FO

Section 4 - Groupings of formatting objects for flow

Those objects represented by the `%block;` parameter entity used in content models:

- `<block>`(6.5.2;81)
  - the formatting specification for a block of lines, distinct from any preceding block of lines
- `<block-container>`(6.5.3;98)
  - the specification of a block-level reference area for contained descendant blocks
- `<list-block>`(6.8.2;87)
  - the parent object of a related set of child list members
- `<table-and-caption>`(6.7.2;93)
  - the parent object of a captioned collection of tabular content
- `<table>`(6.7.3;93)
  - the parent object of an uncaptioned collection of tabular content

Allowed as children of the `<flow>` object

- may also be nested inside of other constructs

Stacked in the block-progression-direction of flow

- these interrupt the block-progression flow of information
- two objects of these types cannot be positioned next to each other in the inline-progression-direction within the same containing object
- note that `<block-container>` can selectively be absolutely positioned outside of the flow and its areas will then not stack with its sibling areas

## Inline-level objects

Chapter 3 - Basic concepts of XSL-FO

Section 4 - Groupings of formatting objects for flow



Those objects represented by the `%inline;` parameter entity used in content models:

- `<basic-link>`(6.9.2;87)
  - the inline content of the start of a unidirectional link to a single end point
- `<bidirectional-override>`(6.6.2;119)
  - overriding the inherent Unicode text direction for a sequence of characters
- `<character />`(6.6.3;119)
  - the inline display of a single character
- `<external-graphic />`(6.6.5;87)
  - the inline display of graphical or other externally-supplied information
- ① `<index-key-reference>`(6.10.6;113)
  - an inline display of resolved page numbers
- `<inline>`(6.6.7;81)
  - the formatting specification for inline content that is distinct from its preceding content within a line generated in a block
- `<inline-container>`(6.6.8;98)
  - an inline reference area for contained blocks
- `<instream-foreign-object>`(6.6.6;87)
  - the inline display of graphical or other instance-supplied information
- `<leader>`(6.6.9;87)
  - the inline display of a rule or a sequence of characters
- `<multi-toggle>`(6.9.5;122)
  - the specification of those interaction-sensitive formatting objects
- `<page-number />`(6.6.10;104)
  - an inline-level place holder replaced with the page number of the current page
- `<page-number-citation />`(6.6.11;81)
  - an inline-level place holder replaced with the page number of the first normal area of the cited formatting object
- ① `<page-number-citation-last />`(6.6.12;81)
  - an inline-level place holder replaced with the page number of the last normal area of the cited formatting object
- ① `<scaling-value-citation />`(6.6.15;87)
  - the inline display of the scaling factor applied to an external graphic

Not allowed as children of the `<flow>` object

- may be nested inside of constructs (including other inline constructs)

Stacked in the inline-progression-direction of flow

- these do not interrupt the block-progression flow of information
- they reside in the line areas generated by the formatter



## Neutral objects

Chapter 3 - Basic concepts of XSL-FO

Section 4 - Groupings of formatting objects for flow

Objects typically allowed anywhere where `#PCDATA`, `%inline;`, or `%block;` constructs are allowed:

- `<change-bar-begin/>`(6.13.2;119)
  - the indication of the start of a change bar
- `<change-bar-end/>`(6.13.3;119)
  - the indication of the end of a change bar
- `<multi-properties>`(6.9.6;122)
  - the collection of candidate property sets from which exactly one set influences the properties of a formatting object based on its status or the status of user interaction
- `<multi-switch>`(6.9.3;122)
  - the collection of candidate formatting object sequences from which exactly one is rendered at any given time based on an interactive condition that is influenced by the operator while being tracked by the formatter
- `<retrieve-marker/>`(6.13.6;104)
  - a place holder replaced with the formatting objects of the indicated marker and allowed only within static content
- `<retrieve-table-marker/>`(6.13.7;104)
  - a place holder replaced with the formatting objects of the indicated marker and allowed only within table header and footer or as a replacement of a table header or footer
- `<wrapper>`(6.13.4;81)
  - a generic container construct for specifying inherited properties for descendent constructs

Note that individual flow objects above may have constraints preventing their use in particular objects.

Areas returned by the interpretation of these objects are stacked in the progression-direction of the siblings of these objects

- except for `<retrieve-marker>` the others may be children of the `<flow>` object

## Out-of-line objects

Chapter 3 - Basic concepts of XSL-FO

Section 4 - Groupings of formatting objects for flow



Out-of-line objects typically allowed anywhere where #PCDATA, %inline;, or %block; constructs are allowed:

- `<float>(6.12.2;98)`
  - content that is to be rendered towards either the before, start, or end edges of a region regardless of where in the region the content is defined

Not stacked in the progression-direction of the sibling objects

- areas returned are contained within and governed by ancestral `<page-sequence>` object

Allowed as a child of the `<flow>` object

- may also be nested inside of other constructs



## Out-of-line inline-level objects

Chapter 3 - Basic concepts of XSL-FO

Section 4 - Groupings of formatting objects for flow

---

Out-of-line objects allowed only inline, i.e.: anywhere where #PCDATA or %inline; constructs are allowed:

- <footnote>(6.12.3;98)
  - content that is to be rendered towards the after-edge of a region regardless of where in the region the content is defined

One of two generated areas stacked in the progression-direction of the sibling inline objects

- other area returned is contained within and governed by ancestral <page-sequence> object

Not allowed as a child of the <flow> object

- must be placed in a block to behave at the block level

## Chapter 4 - Area and page basics



- 
- Introduction - Area and page overview

PREVIEW EXCERPT

# Area and page overview

## Chapter 4 - Area and page basics

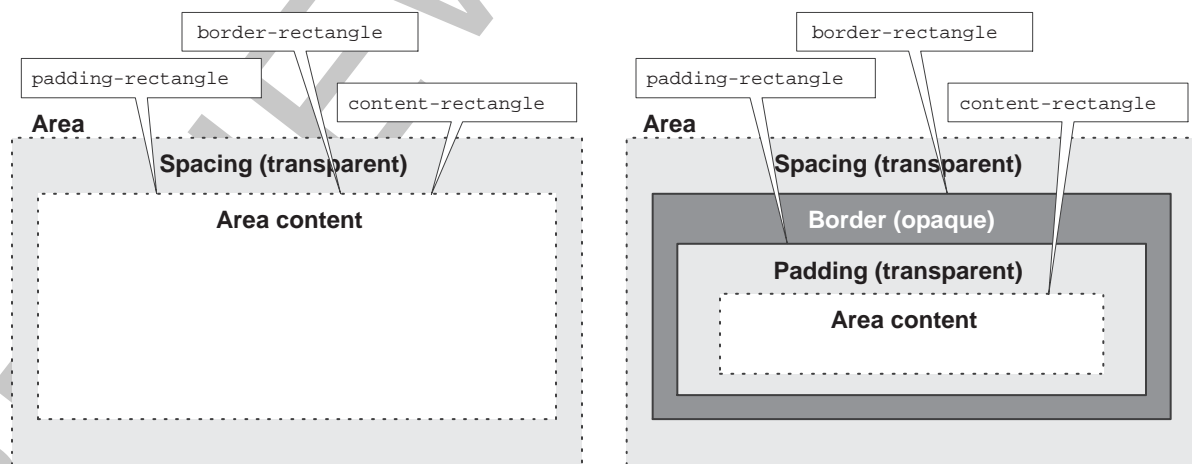


Understanding areas and pages is critical to writing successful XSL-FO instances

- the rendering process renders the area tree created by formatting objects
- to get the desired rendered result, one must know how to position and nest areas of content and set their traits
  - formatting objects are chosen to give the desired layout result
  - the names of formatting objects may be totally irrelevant to the reason they are used in an XSL-FO instance to get the particular areas desired

Area model describes the nature of the areas of content that can be created

- XSL-FO 1 only defines rectangular areas (some of which may be square)
- areas of canvas arranged in hierarchical order in area tree (see page 50)
  - child areas arranged inside the parent area's content-rectangle
  - formatting objects may add areas to multiple branches of the hierarchy
  - objects in a given branch of the area tree stack next to each other
- many different rectangles define the formatter's behavior for the area's content
  - area itself is spaced between its siblings and within parent using a transparent spacing specification
  - rectangles of an opaque border around content may be specified
    - with a border thickness described by the differences between respective edges of two rectangles and a pattern with transparent background
  - rectangles of transparent perimeter spacing around rendered child content within the inside edge of the border (the padding rectangle)
- the number of rectangles in play and their nuances can be overwhelming
  - it is not necessary to know all the rectangles to get simple good-quality results
  - it is important to be aware of the different rectangles to better understand the interplay of areas and the controls available in XSL-FO properties



## Area and page overview (cont.)

Chapter 4 - Area and page basics



---

Writing direction and reference orientation govern visual placement of areas on a page

- these values define the block-progression and inline-progression directions for an area
- supports natural directions of common writing systems of the world
- orientation can be overridden to produce special effects in the rendered result
- values also define the before- and after-sides in the block-progression direction and the start- and end-sides in the inline-progression direction

Areas on the page are not mutually exclusive

- areas can be formatted to overlap other areas in whole or in part
  - transparent backgrounds show other areas behind
- sibling areas in a given branch of the area tree typically do not overlap
- common formatting problems occur when areas from different branches of the tree occupy the same real estate on the page
  - must plan ahead so the stacking of areas in one branch doesn't interfere with the stacking of areas on other branches



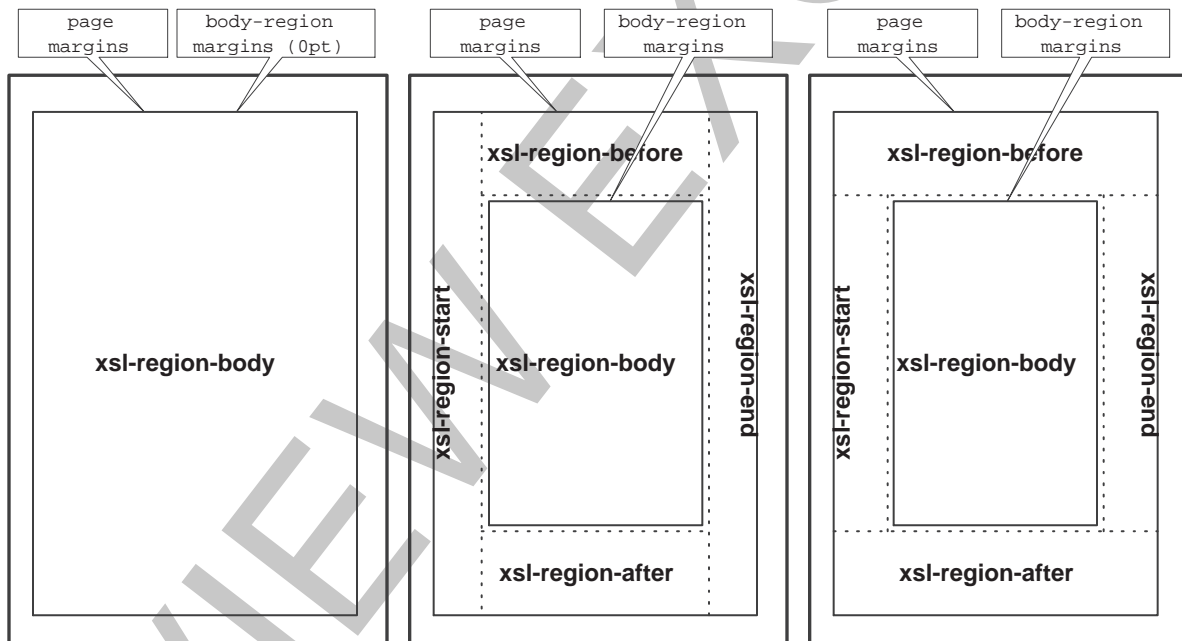
# Page geometry

Chapter 4 - Area and page basics



A page is described by the geometry of its size and various regions

- every page has a `<region-body>` with the initial name `xsl-region-body`
  - content flowed in this region goes is in the main reference area
- possible incursion into the body by four perimeter regions on the four edges
  - `<region-before>` with the initial name `"xsl-region-before"`
  - `<region-after>` with the initial name `"xsl-region-after"`
  - `<region-start>` with the initial name `"xsl-region-start"`
  - `<region-end>` with the initial name `"xsl-region-end"`
  - precedence controls which regions occupy the corners of the perimeter
    - default precedence given to `<region-start>` and `<region-end>`
    - overridden only by `precedence=` property on `<region-before>` and `<region-after>`
- regions are referenced in XSL-FO by using their name
  - regions can be custom-named



Of note:

- the middle page shows the start and end regions have precedence for the corners of the page over the before and after regions (the default)
- the `precedence=` property is used to give precedence individually to the before and after regions when needed

## Page geometry (cont.)

Chapter 4 - Area and page basics



Page regions are targets for either paginated flow or static content

- paginated flow triggers as many pages as needed by the amount of flowed content
  - when a page's region accepting flow overflows, a new page in the page sequence is triggered
  - different regions on separate pages can accept paginated flow (but not on the same page)
  - the flow indicates the name of the target region for the content
- static content for reproduction on every page triggered by pagination
  - the static definition indicates the name of the target region for the content
    - only if the new page includes the named region does a given region's static content get rendered
  - components of static content may be dynamically populated with the page number and the content of user-defined markers appearing on the page being formatted

The formatting objects in each region create descendant areas in that region's branch in the area tree

- the sibling areas in each branch stack separately from the sibling areas in other branches
- without proper body region margins, the perimeter region areas will overlap on top of the body region areas



# Area and page constructs

## Chapter 4 - Area and page basics

The XSL-FO objects covered in this chapter are summarized as follows.

Content-oriented formatting objects:

- `<wrapper>` (6.13.4)
  - a neutral construct for specifying inherited properties for descendent constructs
- `<block>` (6.5.2)
  - the description of canvas content that is distinct from its preceding area content
- `<initial-property-set>` (6.6.4)
  - an auxiliary construct for specifying properties applied to the first line of the parent
- `<inline>` (6.6.7)
  - a description of canvas content that is distinct from its preceding content within a line generated in a block
- `<page-number-citation>` (6.6.11)
  - an inline-level place holder replaced with the page number of the first normal area of the cited formatting object
- `&#106; <page-number-citation-last>` (6.6.12)
  - an inline-level place holder replaced with the page number of the last normal area of a sequence of pages

Page-oriented formatting objects:

- `<simple-page-master>` (6.4.13)
  - the specification of a given page's physical geometry
- `<region-body>` (6.4.14)
  - the definition of the middle area inside any perimeter defined for the page
- `<title>` (6.4.21)
  - a page sequence's ancillary description not rendered on the page canvas

## Chapter 5 - Generic body constructs



- 
- Introduction - Often-used formatting constructs

PREVIEW EXCERPT

# Often-used formatting constructs

## Chapter 5 - Generic body constructs



Many publishing requirements involve the following commonly-used constructs:

- pairs of aligned block-level layout areas in the inline-progression direction
  - normally flowing a block-level area breaks the block progression direction, stacking the block area after the previous area
    - holds two block-level areas stacked beside each other
    - block areas are aligned on their respective before edges
  - coupled members (e.g. side-by-side translated polyglot (multi-language) text)
  - unordered members of a collection (e.g. bulleted lists)
  - ordered members of a collection (e.g. sequenced lists with labels of alpha, roman, number, etc.)
- non-textual information
  - static images or dynamic windows into live applications
  - information external to the XSL-FO instance
    - mandatory for non-XML expressions of the information
    - optional for XML expressions of the information (e.g. SVG)
  - information embedded in the XSL-FO instance
    - must be an XML expression of the information (e.g. SVG)
- unidirectional associations of information
  - from areas of the area tree to a target external resource
  - from areas of the area tree to a target area in the same area tree
  - interactivity engaged by the operator viewing the rendered result can navigate the operator to the target location triggering a traversal of the association
  - unidirectional links have no "back link" information to retrace steps
    - such functionality in browsers and page turner applications is maintained by the application itself, not by the inherent properties of the link
- elastic and inelastic inline areas and decorations
  - forcing inline items to the opposite boundaries of a line
  - assistance when the eye travels from one side of a page to the opposite side
    - used in entries in tables of content
  - patterned sequences of characters joining information on a single line
    - e.g. a dot leader
  - drawn rules in the inline-progression direction
    - a non-textual straight-ruled mark
  - when the length is specified the construct is inelastic
    - useful standalone to break up the flow of information with visual barriers
  - when the length cannot be predetermined, the construct can be elastic
    - grows to the length needed

# Often-used formatting constructs (cont.)

## Chapter 5 - Generic body constructs



An example of a side-by-side bilingual text presentation of an excerpt from the Canadian statute "Employment Equity Act, S.C. 1995, c. 44":

13. Every employer shall, at least once during the period in respect of which the short term numerical goals referred to in paragraph 10(1)(d) are established, review its employment equity plan and revise it by

(a) updating the numerical goals, taking into account the factors referred to in subsection 10 (2); and

(b) making any other changes that are necessary as a result of an assessment made pursuant to paragraph 12(b) or as a result of changing circumstances.

14. Every employer shall provide information to its employees explaining the purpose of employment equity and shall keep its employees informed about measures the employer has undertaken or is planning to undertake to implement employment equity and the progress the employer has made in implementing employment equity.

15. (1) Every employer shall consult with its employees' representatives by inviting the representatives to provide their views concerning

13. Au moins une fois au cours de la période pour laquelle les objectifs quantitatifs à court terme sont fixés, l'employeur procède à la révision de son plan en lui apportant les aménagements rendus nécessaires du fait du suivi ou du changement de sa situation et en adaptant les objectifs quantitatifs, compte tenu des facteurs visés au paragraphe 10(2).

14. L'employeur informe ses salariés sur l'objet de l'équité en matière d'emploi et leur fait part des mesures qu'il a prises ou qu'il entend prendre pour réaliser l'équité en matière d'emploi, ainsi que des progrès qu'il a accomplis dans ce domaine.

15. (1) L'employeur consulte les représentants des salariés et les invite à donner leur avis sur les questions suivantes :

Note:

- the paragraphs are aligned at the before edges
- the sizes of the respective paragraphs are different
- no need to calculate the distance needed to align the starts of paragraphs

# Often-used formatting constructs (cont.)

Chapter 5 - Generic body constructs



Aligned pairs of block-level layout areas can be used for different purposes

- side-by-side presentation
  - e.g. simultaneous translation with aligned paragraphs
- traditional list structures
  - e.g. numbered lists, bulleted lists, terminology definitions, etc.

<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>

The same XSL-FO semantic is used for both kinds of layout above.


# Often-used formatting constructs (cont.)

Chapter 5 - Generic body constructs



The `samp/leadlink.fo` example illustrates leaders, links and graphics:

## The Leader/Link/Graphic Example

Table of Contents	
	
First Title .....	2
Second Title .....	3
Third Title .....	4
<hr/>	
Page count .....	4

The text of each line of the table of contents is "hot"

- the operator interacts with a hot area in order to traverse a link
  - e.g. a mouse click
- the operator is moved to the focus to the target of the association
  - e.g. another page in the same XSL-FO formatted result
  - e.g. a page in the another XSL-FO formatted result
  - e.g. a web browser with a web page address

A graphic image shown below the title

- in this example, this is an external bit image

Various leaders are used on the page

- near the top there are inelastic rule leaders 100% of the width of the page
- above the page count there is an inelastic rule leader 60% the width of the page
- in each entry there is an elastic dot leader that stretches between the start-aligned titles and the end-aligned page numbers





## Often-used formatting constructs (cont.)

Chapter 5 - Generic body constructs

The XSL-FO objects covered in this chapter are summarized as follows.

List objects:

- `<list-block>` (6.8.2)
  - the collection object of a related set of child member pairs of aligned block-level areas
- `<list-item>` (6.8.3)
  - a member pair of aligned block-level areas in a collection
- `<list-item-label>` (6.8.5)
  - the start-side member of a pair of aligned block-level areas
- `<list-item-body>` (6.8.4)
  - the end-side member of a pair of aligned block-level areas

Graphic objects:

- `<external-graphic>` (6.6.5)
  - the inline display of graphical or other externally-supplied information
- `<instream-foreign-object>` (6.6.6)
  - the inline display of graphical or other instance-supplied information
- `<scaling-value-citation>` (6.6.15)
  - the inline display of scaling factor applied to the cited instance-supplied object or externally-supplied information

Link object:

- `<basic-link>` (6.9.2)
  - the inline display of the start resource of a unidirectional link to a single end point

Leader/rule object:

- `<leader>` (6.6.9)
  - the inline elastic or rigid display of a rule or a repeated sequence of characters

## Chapter 6 - Tables



- 
- Introduction - Tabular presentation of information

### Outcomes:

- understand basic concepts of tabular presentation

PREVIEW EXCERPT



# Tabular presentation of information

## Chapter 6 - Tables

A tabular presentation arranges information in a rigid partitioning of groups

- tuples of aligned block-level layout areas
  - see Often-used formatting constructs (page 84) for an example of bilingual text formatting
  - a collection (rows) of collections (columns) of information
  - columns and rows are not being used as indices of a Cartesian plane
    - the grid is only a layout strategy
  - e.g. columns of aligned formatted paragraphs of polyglot (multiple language) text
    - each piece of information is a stand-alone paragraph where each language's paragraph has a different length thus requiring the first line of all translations of the next paragraph to be aligned
- block-level layout areas in a two-dimensional relationship between members of collections
  - traditional Cartesian arrangement of information at the intersection of indexed row and column axes
  - e.g. the corresponding percentages of male and female population at the intersection of age groups in the columns and countries in the rows
    - each piece of information could be a two-line presentation of the percentages for each gender
    - the reader correlates the age group column with the country row to find the information in the corresponding cell

Supports numerous block-level areas arranged in the inline-progression direction

- table contains rows (in block-progression direction) of cells (in inline-progression direction) where each cell contains block-level areas (in block-progression direction)
  - block-level areas would otherwise be arranged in the block-progression direction
- column widths can differ but are fixed for the length of the table
  - widths can be specified values in the supplied objects and properties
  - widths can be based on an automatic weighing of the contents of all of the cells of the table
    - e.g. HTML browser web agents balance column widths based on content

- a collection of tuples of information
  - each member of a tuple in a column
  - each tuple in a row
- a coordinate-based layout of cells of information
  - one ordinate along the columns
  - the other ordinate along the rows
  - the corresponding cells in the main area

Handwriting practice paper featuring a large, light gray 'E' watermark in the background. The page is divided into two main sections by a vertical line. The left section contains a grid of 20 boxes (5 rows by 4 columns) for letter tracing. The right section contains a grid of 20 boxes (5 rows by 4 columns) for letter tracing. Each box is designed for a single letter, with a solid top and bottom line and a dashed middle line for tracing. The boxes are arranged in a 5x4 grid on both sides of the vertical line.

- the column widths are fixed to the same values for all rows in the entire table
  - as in HTML the widths may be based on a balancing of the content of the columns in all of the rows (default)

# Tabular presentation of information (cont.)

## Chapter 6 - Tables



An example of a Cartesian grid presentation of hockey standings:

### Division standings - 2003-04-04

#### Eastern Conference

Northeast Division						
City	Games					Points
	Total	Wins	Losses	Ties	OTL	
Ottawa	81	51	21	8	1	111
Toronto	81	44	27	7	3	98
Boston	81	35	31	11	4	85
Montreal	81	29	35	8	9	75
Buffalo	80	27	36	9	8	71
Atlantic Division						
City	Games					Points
	Total	Wins	Losses	Ties	OTL	
New Jersey	80	45	20	9	6	105
Philadelphia	80	43	20	13	4	103
NY Islanders	79	34	32	11	2	81
NY Rangers	80	32	34	10	4	78
Pittsburgh	81	27	43	6	5	65
Southeast Division						
City	Games					Points
	Total	Wins	Losses	Ties	OTL	
Tampa Bay	80	36	23	16	5	93
Washington	81	38	29	8	6	90
Atlanta	80	29	39	7	5	70
Florida	80	23	35	13	9	68
Carolina	80	22	41	11	6	61

Note:

- rows are spanned for the "Columns" and "Points" headings
- columns are spanned for the "Games" heading

# Tabular presentation of information (cont.)

Chapter 6 - Tables



Independent of the organization of source information

- the original data needn't be modeled by a table construct
  - historical use of structured information tools often forced information designers to model explicit table constructs
  - such models insufficiently capture the semantics of the information being related
- information from any model can be presented in a tabular fashion
  - transformation can rearrange the source information into the tabular relationship

An XSL-FO table has a compound structure with many well-defined behaviors

- a block-level construct that breaks the flow in the block-progression direction
- the header rows and footer rows are constructs repeated on all pages where the table's body rows are rendered
  - the caption is only shown on the first page containing part of the table
- column properties can be specified once for all cells of rows that are in a given column
  - the linear serialization of a two-dimensional construct necessarily favors one dimension over the other
  - cells are contained within rows, not columns, so a column-oriented construct is necessary to address the second dimension of property assignment
- column-spanning and row-spanning features provide flexible table cell definition



# Tabular presentation of information (cont.)

## Chapter 6 - Tables

The XSL-FO objects covered in this chapter are:

- `<table-and-caption>` (6.7.2)
  - the parent object of a captioned collection of tabular content
- `<table-caption>` (6.7.5)
  - the caption of a captioned collection of tabular content
- `<table>` (6.7.3)
  - the parent object of an uncaptioned collection of tabular content
- `<table-column>` (6.7.4)
  - the specification of common columnar properties
- `<table-header>` (6.7.6)
  - the rows of tabular content repeated at the before-edge of every break in body content
- `<table-footer>` (6.7.7)
  - the rows of tabular content repeated at the after-edge of every break in body content
- `<table-body>` (6.7.8)
  - the rows of tabular content flowed as the body content
- `<table-row>` (6.7.9)
  - a row of tabular content
- `<table-cell>` (6.7.10)
  - a column of a row of tabular content

## Chapter 7 - Floats, footnotes and containers



- 
- Introduction - Out-of-line publishing constructs

### Outcomes:

- understand basic concepts of out-of-line constructs

PREVIEW EXCERPT





# Out-of-line publishing constructs

Chapter 7 - Floats, footnotes and containers

Floats, footnotes and absolutely positioned containers render information supplemental to the information found in the flow:

- information is easily found by a reader because of a predetermined location on the page
  - defined "in line" of the flow of information being paginated
  - rendered "out-of-line" of the flow of information being paginated
  - considered auxiliary enough not to disturb the flow itself for the reader
    - reader can choose to examine a float or footnote at leisure without interrupting the reading of the flow
- dynamically rendered on the page where detected by the formatter in the flow
  - can only be defined in the flow filling the body region of a page
    - also cannot be defined in an absolutely positioned area
  - it is stacked in a different reference area than the main reference area
- floats are rendered to either the before, start, or end edges within the body region
  - not in the perimeter regions
- footnotes are two-part constructs:
  - footnote citation rendered inline in the flow
  - footnote body rendered at the after edge of the body region
  - an after float is accomplished using a footnote without a footnote citation
    - only when footnotes are not already being used
- block containers can be absolutely positioned on the page
  - relative to the parent or relative to the medium
  - inline containers are positioned inline but have the same block-oriented content as a block container
  - the construct is named by where it is allowed to be specified, not by its contents
- out-of-line objects may not have out-of-line descendent objects

Footnote labeling is the responsibility of transformation

- the label of a footnote reference is supplied in the instance and not generated by the formatter
- restarting the footnote referencing on a per-page basis is not available in XSL-FO 1.0

No endnote constructs in XSL-FO

- responsibility of transformation to implement endnote functionality
- cite endnotes inline in the flow of the scope
- collect and render endnotes at the end of scope
  - can use empty citations at end to take advantage of XSL-FO footnote construct

## Out-of-line publishing constructs (cont.)

Chapter 7 - Floats, footnotes and containers



Using out-of-line constructs allows the normal flow to stack in the body without interruption

- floats stack at the perimeter edges of the page
- can prevent large gaps in the normal flow when large constructs would otherwise jump to a new page

Many candidate uses of out-of-line constructs

- using `<float>` either within a block or between blocks:
  - floating images to the top of a page
  - side-bar presentations
  - lists with item bodies indented relative to the corresponding item label's formatted length
  - a multi-line drop initial cap in a paragraph
- using `<footnote>` allowed only within a block:
  - traditional footnotes
  - acronym expansions
  - glossary definitions
  - sinking images to the bottom of a page (using an empty inline construct)
  - trailing disclaimer at the bottom of the last page of a rendered document
- using absolutely-positioned `<block-container>` allowed either within a block or between blocks
  - positioning a logo in static content
  - positioning crop marks in an oversized page

The name of the construct shouldn't prejudice how the construct is used

- consider the need mentioned above to format a disclaimer at the bottom of the last page of a document
- one can flow the disclaimer in the body of a footnote with an empty citation in an empty block at the end of the document

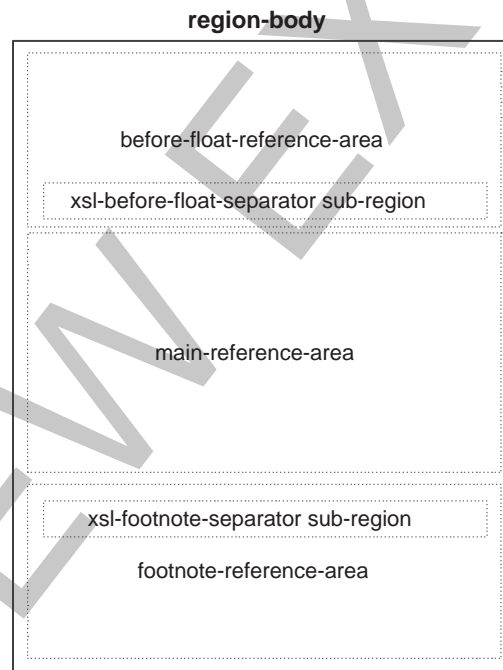
# Out-of-line publishing constructs (cont.)

Chapter 7 - Floats, footnotes and containers



Every page's body region has two sub-regions rendered only if necessary

- before-floats and footnotes are stacked in the body region with other block-level constructs
  - the reader needs some separation rendered to distinguish content belonging in a float or footnote from the content belonging in the body
  - the body region is separated into the before-float-reference-area, main-reference-area, and footnote-reference-area
- the act of defining the separators does not effect their rendering
  - defined using static content
  - only rendered on a page if the floated information is being rendered on the page
  - `xsl-before-float-separator`
    - inside and at the end of the before-float-reference-area
  - `xsl-footnote-separator`
    - inside and at the start of the footnote-reference-area
- should always be defined as a contingency if floats and footnotes are being used



Remember from Page geometry (page 79) the incursion of perimeter regions into the body if the body margins are not set accordingly

- all reference areas shown above are within the body-region's margins

## Out-of-line publishing constructs (cont.)

Chapter 7 - Floats, footnotes and containers



---

The XSL-FO objects covered in this chapter are:

- `<float>` (6.12.2)
  - content that is to be rendered towards either the before, start, or end edges of the body region regardless of where in the region the content is defined
- `<footnote>` (6.12.3)
  - content that is to be rendered both in the flow and towards the after-edge of the body region regardless of where in the region the content is defined
- `<footnote-body>` (6.12.4)
  - the portion of footnote content rendered towards the after-edge of the body region
- `<block-container>` (6.5.3)
  - the specification of a block-level reference area for contained descendant blocks
- `<inline-container>` (6.6.8)
  - the specification of an inline-level reference area for contained descendant blocks

## Chapter 8 - Flows, static content and page geometry sequencing



- 
- Introduction - Extended page model for pagination
  - Section 1 - Page Sequence Master Interleave (PSMI)

### Outcomes:

- understand basic concepts of pagination constructs

PREVIEW EXCERPT

# Extended page model for pagination

Chapter 8 - Flows, static content and page geometry sequencing



Bounded areas (pages) repeat to accommodate an arbitrary amount of content (flow)

- pagination differentiates XSL-FO semantics from web browser display semantics

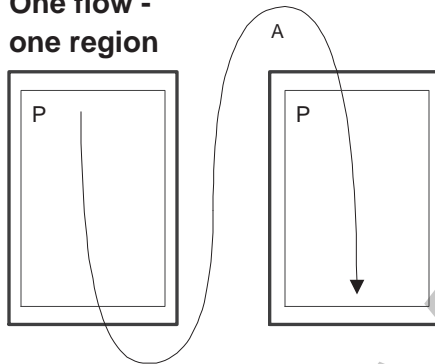
## 1 Exactly one paginating flow is defined in XSL-FO 1.0

- an implicit flow map corresponds flow names to region names
- only serpentine flow is through equal-width evenly-spaced columns

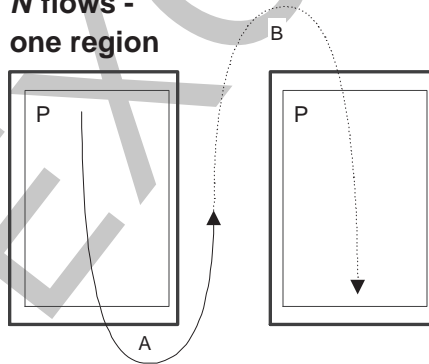
## 1 One or more paginating flows can be defined in XSL-FO 1.1

- a stylesheet-defined flow map arbitrarily corresponds flow names to region names for desired effects
- the same 1.0 implicit flow map corresponds flow names to region names when the stylesheet does not provide an explicit flow map
- multiple body regions allow the stylesheet to place many regions arbitrarily on the page

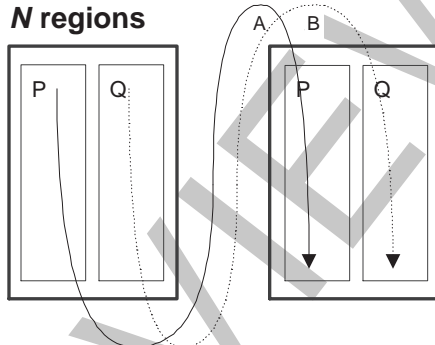
**One flow -  
one region**



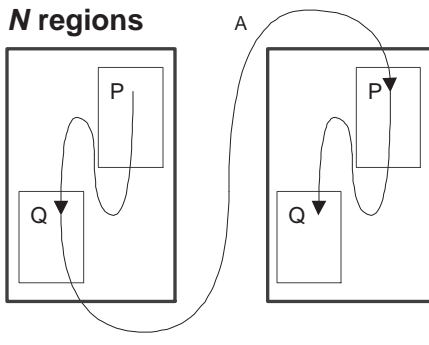
**N flows -  
one region**



**N flows -  
N regions**



**One flow -  
N regions**





## Extended page model for pagination (cont.)

Chapter 8 - Flows, static content and page geometry sequencing

The pages in the collection need navigational aids repeated on each page

- used to understand a page's role when not in the context of neighboring pages
- page numbers and page number citations
  - a focus on the specific bounded area of a piece of information
  - used on the given page for navigation
  - used on other pages for citations to the given page
    - e.g. cross references
    - e.g. index information
- headers and footers
  - contextual information about a collection of pages in which the page is found
  - e.g. chapter title repeated in the header
- cited information found from the page being formatted can be contextual information
  - information that would not be known about a page by the stylesheet at the time of transformation, possibly changing for each page generated
    - the act of pagination dictates the page boundaries, not the transformation of the source information
  - e.g. dictionary headers (finding one of many items on a page)
  - e.g. subsection citations (finding breaks not triggering new pages)

Every page sequence starts a new page and has its own definitions for static content

- all candidate uses of static content must be defined for each page sequence
- necessary for the stylesheet writer to repeat the definitions in each page sequence if the same behavior is desired
- every page sequence begins on a new page
  - changing a header or footer in the middle of a page sequence requires manipulating the dynamic content that can be placed within static content

Static content and paginated flow are associated with a region's name, not position

- association of the flow name through the flow map to the region name
- must organize the desired region names in each page geometry
- must bind in each page sequence the static content and flows for named regions
- all flows are used for only the named regions found on each page geometry used
  - it is not an error to supply static content or flow for a region that isn't used
- static content can be supplied for named sub-regions triggered by the formatter

# Extended page model for pagination (cont.)

Chapter 8 - Flows, static content and page geometry sequencing



Differences in page geometry are allowed when more than one page being rendered

- supports tests for differences to be performed within each page sequence construct's flow when the flow triggers the need for a new page
- different named geometries can be identical to others or have differences such as:
  - the change of physical dimensions
  - the choice of region dimensions and names
  - the change of margins
  - the presence of headers and/or footers
  - the count of columns
  - the orientation of regions
  - the backgrounds of regions
  - the absence of flowed content (to utilize an entire page of static content)

Can describe a sequence with odd and even page number parity differences

- e.g. alternating headers and footers
- two different static contents are defined with the page number to be rendered on the outside edge of each side of a bound publication
  - differences in headers and footers of the geometries alternate the names of the flows for the static content between odd and even numbered pages

Can describe a sequence with first, last, and middle page differences

- to utilize differences based on where a given formatted page is within its containing page sequence
- e.g. no heading on the first page of a chapter sequence

Can describe a sequence to replace absent content for forced un-flowed blank pages

- to accommodate a requirement for a specified parity of pages in a given sequence, or the need for the subsequent page sequence to start on a page following the page after the end of a page sequence's flow
- i.e. "this page intentionally left blank"
- by definition, a forced page is made up entirely of static content

Each page sequence points to a defined sequence of page geometries

- the stylesheet writer choreographs the page geometries by the anticipated needs for the quantity of flow expected to be formatted
- all static content definitions needed for a page sequence must be present in that sequence
  - often requires stylesheet writing techniques that copy information repeatedly



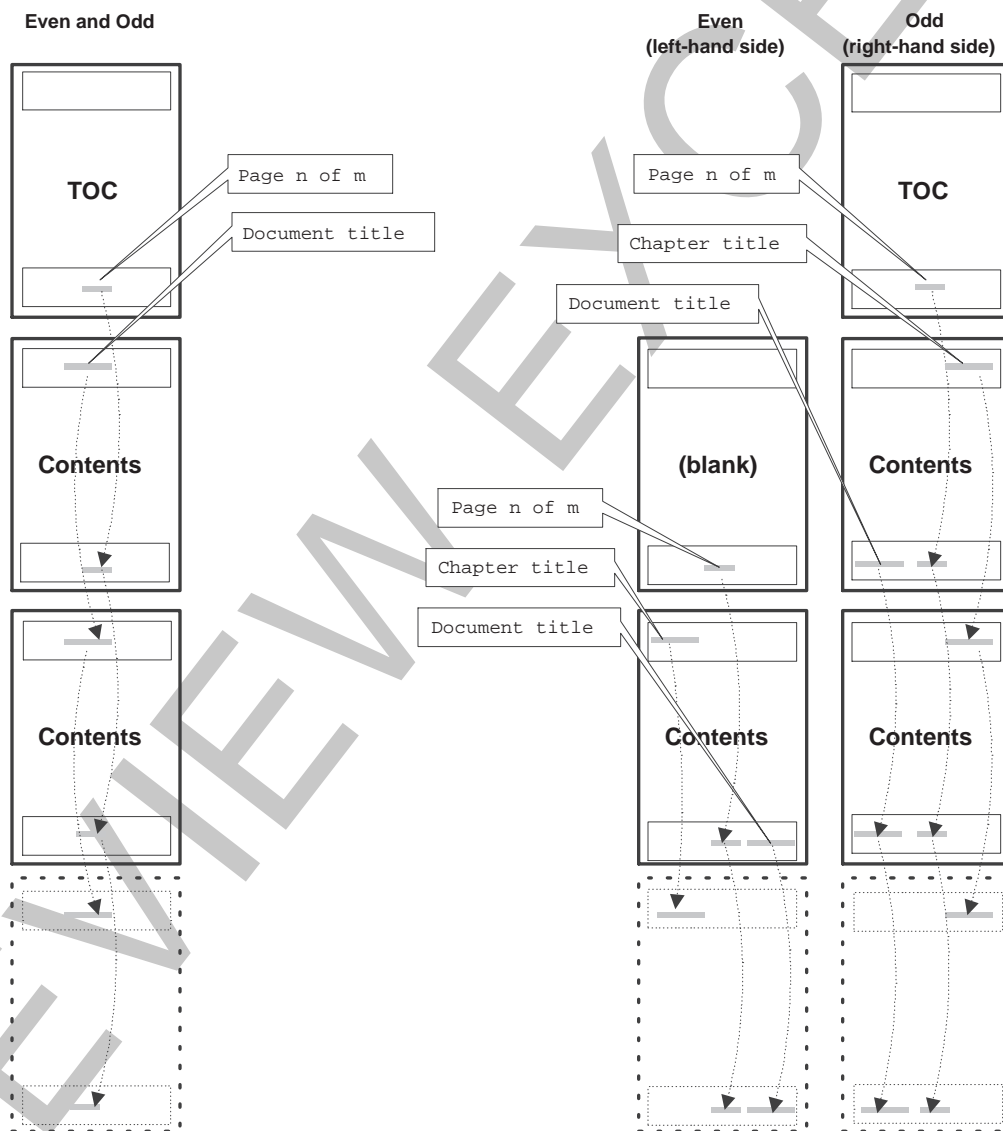
# Extended page model for pagination (cont.)

Chapter 8 - Flows, static content and page geometry sequencing



Consider the choreography in each of two possible page plans for a TOC and chapters:

- a single-sided presentation on the left with all page sequences consecutive:
  - document title centered at the top of document content pages (but not the TOC)
  - page number and total page count centered at the bottom of all pages
- a double-sided presentation on the right with contents starting on right-hand page:
  - requires the table of contents to be an even number of pages
  - document title at bottom left of odd pages and bottom right of even pages
  - chapter title at top right of odd pages and top left of even pages
  - page number and total page count centered at the bottom of all pages



# Extended page model for pagination (cont.)

Chapter 8 - Flows, static content and page geometry sequencing



The XSL-FO objects covered in this chapter are summarized as follows.

Formatting objects related to static content:

- `<region-before>` (6.4.15)
  - the definition of the body region perimeter area whose before-edge is co-incident with the before-edge of the page's content rectangle
    - in `lr-tb` mode this is the header at the top of the page
- `<region-after>` (6.4.16)
  - the definition of the body region perimeter area whose after-edge is co-incident with the after-edge of the page's content rectangle
    - in `lr-tb` mode this is the footer at the bottom of the page
- `<region-start>` (6.4.17)
  - the definition of the body region perimeter area whose start-edge is co-incident with the start-edge of the page's content rectangle
    - in `lr-tb` mode this is the sidebar at the left of the page
- `<region-end>` (6.4.18)
  - the definition of the body region perimeter area whose end-edge is co-incident with the end-edge of the page's content rectangle
    - in `lr-tb` mode this is the sidebar at the right of the page
- `<static-content>` (6.4.20)
  - the definition of content that is primarily unchanged from page to page in a page sequence
    - entire sequence is repeated on each page except for page numbers and user-defined markers
- `<page-number>` (6.6.10)
  - an inline-level place holder replaced with the page number of the current page
- ① `<folio-prefix>` (6.6.13)
  - a sequence of text defined to prefix page numbers of a page sequence
- ① `<folio-suffix>` (6.6.14)
  - a sequence of text defined to suffix page numbers of a page sequence
- `<retrieve-marker>` (6.13.6)
  - an inline-level place holder for perimeter regions, replaced with the formatting objects of the indicated marker
- ① `<retrieve-table-marker>` (6.13.7)
  - an inline-level place holder for tables, replaced with the formatting objects of the indicated marker
- `<marker>` (6.13.5)
  - the replacement formatting object content for a marker retrieved in static content



## Extended page model for pagination (cont.)

Chapter 8 - Flows, static content and page geometry sequencing

Formatting objects related to page geometry sequencing:

- `<page-sequence-master>` (6.4.8)
  - the definition and name of a particular sequence of using page-masters
- `<single-page-master-reference>` (6.4.9)
  - the specification of the single use of a page-master within a sequence of page-masters
- `<repeatable-page-master-reference>` (6.4.10)
  - the specification of the repeated use of a page-master within a sequence of page-masters
- `<repeatable-page-master-alternatives>` (6.4.11)
  - the collection of possible page-master references from which one is to be used based on status conditions detected by the formatter
- `<conditional-page-master-reference>` (6.4.12)
  - a page-master choice available to the formatter when selecting from a collection of candidate page-masters

Formatting objects related to flow maps:

- ① `<flow-map>` (6.4.22)
  - specifies a collection of assignments of a set of named flows to a set of named regions
- ① `<flow-assignment>` (6.4.23)
  - specifies a single assignment of a set of named flows to a set of named regions
- ① `<flow-source-list>` (6.4.24)
  - specifies a set of named flows in a single flow assignment
- ① `<flow-name-specifier>` (6.4.25)
  - specifies a single named flow in a set of named flows
- ① `<flow-target-list>` (6.4.26)
  - specifies a set of named regions in a single flow assignment
- ① `<region-name-specifier>` (6.4.27)
  - specifies a single named region in a set of named regions

# Changing the page geometry based on authored content

Chapter 8 - Flows, static content and page geometry sequencing  
Section 1 - Page Sequence Master Interleave (PSMI)



Sometimes it is necessary to change page geometry based on content

- the XML source author may arbitrarily request a change in the page geometry
- consider the need to change the geometry for a lengthy landscaped table
  - cannot use a rotated table because the table length is limited by the container
  - must switch to a landscape body region so the lengthy table flows to subsequent pages
- consider the need to flow certain figures into double-sized pages
  - must selectively choose a geometry with the different dimensions
- the construct may be deep within the structure of the document being processed

To use XSL-FO one must package all content for each page geometry for the flow of a page sequence

- this can involve a difficult and wasteful recursive process
  - finding all information up to the change in geometry
  - package the information up in a page sequence
  - package up the special constructs in a different page sequence
  - recursively find all information up to the next change in geometry
- this breaks the XSLT model of hierarchical processing
  - susceptible to development and maintenance problems

A two-step process can make this very easy to implement

- the first step creates the flow with supplemental indications of the need to change geometries
  - can come from any depth of processing of the source document
  - block level constructs of a page are the immediate children of the flow
- the second step repackages flow children in as many sequences as necessary
  - recursively checks all flow child constructs for any changes in geometry
  - not necessary to check any depth deeper than the children
- the resulting document is then processed by a standard XSL-FO engine

The Page Sequence Master Interleave (PSMI) semantic implements this algorithm

- public resource freely available from Crane Softwrights Ltd.
  - <http://www.CraneSoftwrights.com/links/res-pfux.htm>
- one-element vocabulary designed to express the semantic of this two-step intermediate process
- an XSLT stylesheet to implement the semantic for any XSL-FO+PSMI sequence

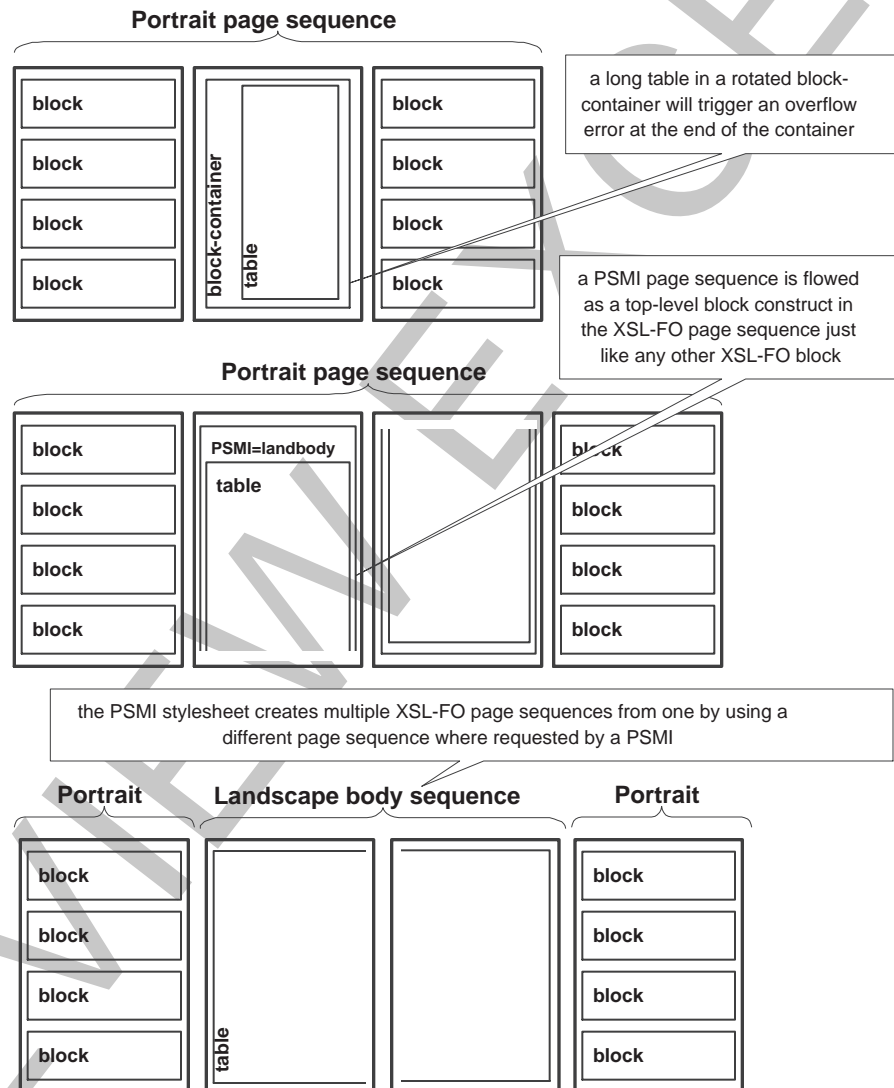
# Changing the page geometry based on authored content (cont.)

Chapter 8 - Flows, static content and page geometry sequencing  
Section 1 - Page Sequence Master Interleave (PSMI)



Consider the need to flow a landscape table in a portrait page geometry

- could rotate a short table in a block container to present in landscape
  - a long table would overflow and not trigger a new page in pagination
- using PSMI one could flow the landscape table into an interleaved page geometry where the body region of that geometry is landscape
- the PSMI stylesheet reads the XSL-FO+PSMI instance and produces a pure XSL-FO instance where three page sequences are created where one was before
- a long table in the landscape body region will correctly trigger new pages in pagination



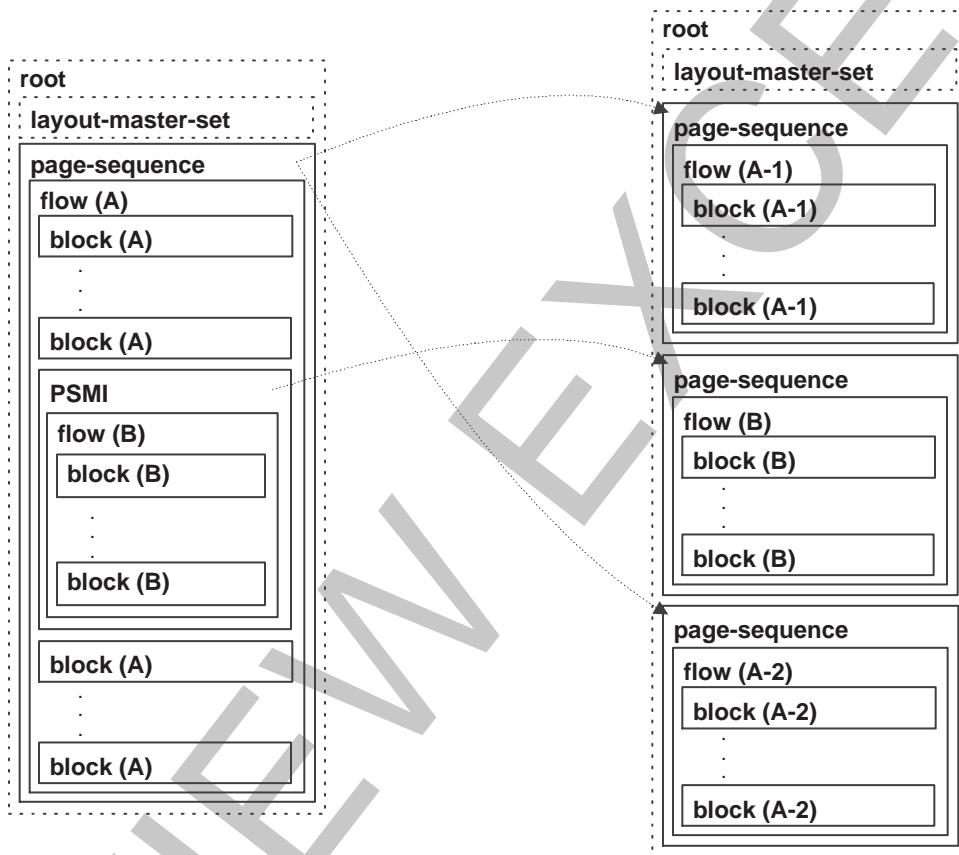
# Changing the page geometry based on authored content (cont.)

Chapter 8 - Flows, static content and page geometry sequencing  
Section 1 - Page Sequence Master Interleave (PSMI)



The end result of PSMI transformation is a pure XSL-FO instance:

- the parent XSL-FO page-sequence of the PSMI page-sequence is split
- a new XSL-FO page-sequence is created for the PSMI page-sequence and contains its blocks
- the preceding siblings of the PSMI page-sequence are put in the preceding page sequence
- the following siblings of the PSMI page-sequence are put in the following page sequence
- the process accommodates any number of PSMI children of an XSL-FO page-sequence



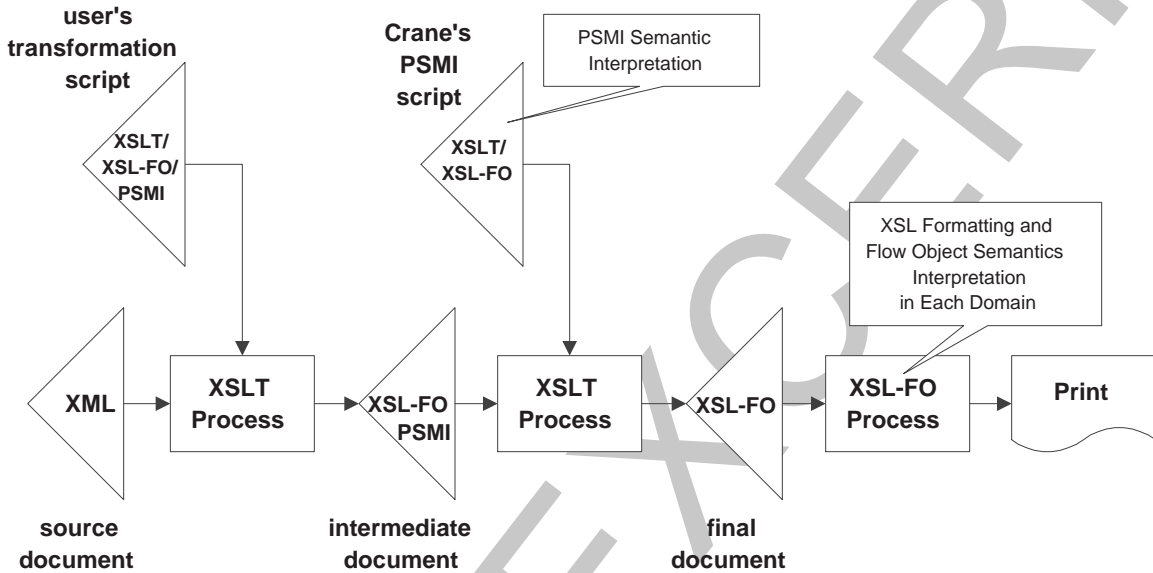


# Changing the page geometry based on authored content (cont.)

Chapter 8 - Flows, static content and page geometry sequencing  
Section 1 - Page Sequence Master Interleave (PSMI)

The two-step process involves:

- the user's stylesheet to produce the XSL-FO+PSMI intermediate sequence
- the PSMI stylesheet to produce a pure XSL-FO instance





# Changing the page geometry based on authored content (cont.)

Chapter 8 - Flows, static content and page geometry sequencing  
Section 1 - Page Sequence Master Interleave (PSMI)



An example of flow implementing the change in page geometry:

```

01    <simple-page-master master-name="frame-portraitbody" ...>
02        <region-body region-name="frame-body" .../>
03        <region-before region-name="frame-before" .../>
04        <region-after region-name="frame-after" .../>
05    </simple-page-master>
06    <simple-page-master master-name="frame-landbody" ...>
07        <region-body region-name="frame-body"
08            reference-orientation="90" .../>
09        <region-before region-name="frame-before" .../>
10        <region-after region-name="frame-after" .../>
11    </simple-page-master>
12 ...
13 <page-sequence master-reference="frame-portraitbody">
14     <flow flow-name="frame-body">
15         <block>Portrait information</block>
16     ...
17     <block>Next is landscaped</block>
18 <psmi:page-sequence master-reference="frame-landbody"
19     xmlns:psmi="http://www.CraneSoftwrights.com/resources/psmi">
20     <flow flow-name="frame-body">
21         <table>
22             <table-body>
23                 <table-row>
24                     <table-cell border="solid">
25                         <block>This is a test</block>
26                         <block>This is a test</block>
27                     ...
28                     <block>This is a test</block>
29                     <block>This is a test</block>
30                 </table-cell>
31             </table-row>
32         </table-body>
33     </table>
34     </flow>
35 </psmi:page-sequence>
36 <block>Back to portrait</block>
37 ...
38 </flow>

```



## Chapter 9 - Bookmarks and indexes



- 
- Introduction - Additional navigation aids

### Outcomes:

- understand basic concepts of bookmark and index formatting objects

PREVIEW EXCERPT

# Additional navigation aids

Chapter 9 - Bookmarks and indexes



Additional navigation aids can help the reader find information easily

- provides hyperlinks into contents

## 1 Bookmarks

- similar to a table of contents, but not presented on the canvas of the publication
  - it is a design pattern to use the same logic to build both the rendered table of contents and the publications bookmark information
- document rendering formats such as Adobe Portable Document Format (PDF) have a bookmark tree
  - viewers render the bookmark tree in a separate window pane
  - interactions in the bookmark pane change the focus of the publication in the main pane
- XSL-FO doesn't apply any presentation constraints or many features on bookmark entries
  - entries are comprised of text and a very small subset of annotations and styles

## 1 Indexes

- far more granular than a table of contents
- a hierarchical alphabetized summary of key concepts and words found in the publication
- using page number citations is insufficient to the task
  - page number citations cannot have page ranges and duplicate page number citations resolved into a minimum list of unique page references
- XSL-FO only addresses page number resolution for indexes, not the index structure or rendering
  - up to the stylesheet to accomplish the selection, grouping and ordering of the index entries

## 1 A two-pass methodology for PDF files can be used with XSLT 1.0

- public resource freely available from Crane Softwrights Ltd.
  - <http://www.CraneSoftwrights.com/links/res-pfux.htm>
- the method first renders XML markup on PDF pages using page number citations to resolve all index entries
- the markup is scraped out of the PDF file by off-the-shelf PDF software
- the markup is massaged by XSLT to interpret page ranges and eliminate duplication
- the massaged markup is input to the second pass to render the index hyperlinked into the content

## 1 Many vendors provide custom extensions for this functionality

- but extensions are not portable across different tools, thus requiring duplication of the information in an instance enabled for different processors

## Additional navigation aids (cont.)

Chapter 9 - Bookmarks and indexes



The XSL-FO objects for bookmarks covered in this chapter are:

- ¶ `<bookmark-tree>` (6.11.1)
  - defines a set of bookmarks for the completed document
- ¶ `<bookmark>` (6.11.2)
  - defines a single bookmark
- ¶ `<bookmark-title>` (6.11.3)
  - specifies the title for a single bookmark

The XSL-FO objects for indexes covered in this chapter are:

- ¶ `<index-page-citation-list>` (6.10.7)
  - render a formatted list of page numbers and page ranges for a set of index keys
- ¶ `<index-page-citation-list-separator>` (6.10.8)
  - specifies the separation between formatted index entry members
- ¶ `<index-page-citation-range-separator>` (6.10.9)
  - specifies the separation between two page numbers in a page range in a formatted index entry member
- ¶ `<index-key-reference>` (6.10.6)
  - render a formatted list of page numbers and page ranges for a single index key
- ¶ `<index-page-number-prefix>` (6.10.2)
  - specifies the start of pages in the flow for an index page range
- ¶ `<index-page-number-suffix>` (6.10.3)
  - specifies the end of pages in the flow for an index page range
- ¶ `<index-range-begin>` (6.10.4)
  - specifies the suffix for cited pages for a single index key
- ¶ `<index-range-end>` (6.10.5)
  - specifies the suffix for cited pages for a single index key

## Chapter 10 - Breaks, keeps, spacing, borders and backgrounds



- 
- Introduction - Spacing and arrangement constraint definition

### Outcomes:

- understand basic concepts of spacing and stacking

PREVIEW EXCERPT

# Spacing and arrangement constraint definition

Chapter 10 - Breaks, keeps, spacing, borders and backgrounds



An area's placement is governed by XSL-FO stacking rules and the area's traits refined from object properties

- at the block level in a page
  - before-float-reference-area
  - normal-flow-reference-areas (page columns)
  - footnote-reference-area
- at the line level in a block
  - lines are generated by the formatter, not the stylesheet
  - different line-stacking strategies are available to be specified
- at an inline-level in a line
  - characters, graphics, etc.

Areas that are stacked normally are stacked in the pertinent progression direction

- page-level reference areas stack in the block-progression direction
- lines and table rows stack in the block-progression direction
- page column, table column and inline areas stack in the inline-progression direction

The "natural" stacking of areas may produce typographically unpleasant results

- initial values implement common-sense formatting control for consistent presentation
- many traditional conventions break the behavior of the initial values
  - e.g. needing to keep a heading in the body on the same page as the first paragraph to which the heading applies when the naturally occurring page break would otherwise come between the two items
  - necessary sometimes to override the physical arrangement of information implied by the initial values for area properties

# Spacing and arrangement constraint definition (cont.)

Chapter 10 - Breaks, keeps, spacing, borders and backgrounds



Conditionality and precedence can eliminate areas from being rendered

- can prevent the unnecessary use of inter-block spacing when pagination renders a block without an adjacent sibling
  - e.g. a block forced to the top of a new page doesn't always need the space defined for between blocks to be rendered
- formatting special cases can be accommodated with simple specifications of intent
  - the formatter determine the applicability of spaces based on object properties
  - the transformation process can ascribe properties easier than determining space behaviors

Numerous properties are available to specify these nuances of layout

- arbitrarily breaking the flow of content to a new column or page
- maintaining a minimum number of widow and orphan lines of a block on a page
- keeping information together in the same reference area
- drawing borders around information
- painting backgrounds behind information

## Chapter 11 - Supplemental objects



- 
- Introduction - Specialty constructs

### Outcomes:

- understand basic concepts of lesser-used constructs

PREVIEW EXCERPT

# Specialty constructs

Chapter 11 - Supplemental objects



The objects described here are not typically used very often but can provide useful functionality to those with specific formatting requirements in three areas:

- ① change bars
  - lines associated with the flow typically used to indicate changed content
- character-level processing outside of the default behaviors
  - managing the bi-directionality inherent in the Unicode properties of characters
  - instantiating formatting objects explicitly in the formatting object tree
- constructs of a global nature
  - specifying a color system recognized by an XSL-FO processor
  - accommodating a home in the XSL-FO instance for color system specifications





## Specialty constructs (cont.)

Chapter 11 - Supplemental objects

---

The XSL-FO objects covered in this chapter are:

- ¶ <change-bar-begin> (6.13.2)
  - indication of the starting of change bar rendering
- ¶ <change-bar-end> (6.13.3)
  - indication of the completion of change bar rendering
- <color-profile> (6.4.4)
  - the declaration of a profile of candidate color values from which color specifications can be made by formatting objects
- <declarations> (6.4.3)
  - a global-scope repository of formatting object constructs for an XSL-FO instance
    - the collection of available color profiles
    - any collections of extended formatting objects supported by an XSL-FO processor
      - must use a processor-recognized namespace URI other than the XSL-FO namespace URI
- <bidirectional-override> (6.6.2)
  - specifies how to manage and override the inherent Unicode text direction for a sequence of characters
- <character> (6.6.3)
  - both the abstract formatting object implied by a simple character in an XSL-FO instance and the concrete formatting object available to be used in place of a simple character

## Chapter 12 - Interactive objects



- 
- Introduction - Dynamic and interactive object rendering

Outcomes:

- understand basic concepts of interactive objects

PREVIEW EXCERPT

# Dynamic and interactive object rendering

Chapter 12 - Interactive objects



---

An electronic canvas can offer an active presentation of content

- operator interaction can influence presentation
  - the use of interactive formatting objects can equip the operator to present the content as desired from multiple alternatives
  - the state of an interactive object can be reflected based on previous interaction or lack thereof
- dynamically changing presentation
  - not restricted to a single static rendering of the information

Two areas where interactivity can influence presentation

- reflecting the active state of a linked object using different property values
  - a link can be but hasn't yet been traversed (future potential for visitation)
  - a link would be traversed (under a hovering pointer)
  - a link that is about to be traversed (has the user interface focus)
  - a link that is in the process of being traversed (is active)
  - a link has been traversed (past visitation)
- selecting and switching between alternate available presentations using different sub-trees of formatting objects
  - e.g. a dynamically expandable and collapsible table of contents rendering
  - HTML pages accomplish this in an imperative fashion with scripting in a programming language
  - XSL-FO 1.0 pages accomplish this in a declarative fashion by describing transition paths in a state machine, without having to implement the imperative logic behind the state machine

# Dynamic and interactive object rendering (cont.)

Chapter 12 - Interactive objects



Formatting objects related to dynamic properties are:

- `<multi-properties>` (6.9.6)
  - the collection of candidate property sets from which exactly one set influences the properties of a formatting object based on its status or the status of operator interaction
- `<multi-property-set>` (6.9.7)
  - the set of properties associated with a single possible state of a formatting object or operator interaction

Formatting objects related to dynamic presentation are:

- `<multi-switch>` (6.9.3)
  - the collection of candidate formatting object sequences from which exactly one is rendered at any given time based on an interactive condition that is influenced by the operator while being tracked by the formatter
- `<multi-case>` (6.9.4)
  - a single formatting object sequence that is a candidate for rendering based on an interactive condition that is influenced by the operator while being tracked by the formatter
- `<multi-toggle>` (6.9.5)
  - the definition of those interaction-sensitive objects within a candidate rendered sequence of formatting objects

## Chapter 13 - Where XSL-FO 1 falls short



- 
- Introduction - What is missing in XSL-FO 1?

PREVIEW EXCERPT

# What is missing in XSL-FO 1?

Chapter 13 - Where XSL-FO 1 falls short



Important note: this section is conjecture on the part of the author and does not represent any official (or even unofficial) comments from the development committee.

Of course the designers of XSL-FO 1 could not put everything into the specification and ever consider their work "complete", but what is available is very useful

- measured amount of functionality to ensure successful implementation and deployment
- need real-world experience with first definition to ensure design principles are sound
- a number of features are already addressed in XSL-FO 1.1

Indications are in XSL-FO 1 of more complex functionality to follow

- the `<simple-page-master>` construct name implies perhaps a more complete page geometry definition is planned for the future
- property set collections of only a single property
  - if not planning for the future, why not just list the property instead of putting it alone in a set?

Some important functionality needed by compositors is left to a future version

- page-related formatting requirements
- geometry-related formatting requirements
- alignment of areas

Many vendors offer a wide range of extra features

- new nuance
- new functionality

## Annex A - Using XSLT with XSL-FO



- 
- Introduction - Using XSLT with XSL-FO
  - Section 1 - XSLT language features supporting XSL-FO
  - Section 2 - XSL-FO language features similar to XSLT and XPath

PREVIEW EXCERPT

# Using XSLT with XSL-FO

Annex A - Using XSLT with XSL-FO



XSLT was designed primarily for use with XSL-FO

- result of an XSLT transform is an XML information set that may or may not be serialized as XML syntax
  - result information set can be delivered to XSL-FO formatter without serialization
- a number of features of XSLT assist the writing of XSL-FO stylesheets
- the design of XSL-FO is made simpler by the use of XSLT functionality

XSLT is a normative part of XSL

- section 2.1 of XSL states the following:
  - The provisions in "XSL Transformations" form an integral part of this recommendation and are considered normative.

Certain aspects of XSL-FO are defined by XSLT

- references are made in XSL-FO to sections of XSLT for their definition
- design decisions in XSL-FO pattern themselves after XSLT definitions

No technical reason that XSLT must be used to create XSL-FO

- the formatter can accept an instance of XSL-FO regardless of how that instance was generated



## <xsl:attribute-set> instruction

Annex A - Using XSLT with XSL-FO

Section 1 - XSLT language features supporting XSL-FO



<xsl:attribute-set> is a very useful instruction for the manipulation of many attributes targeted for a given formatting object:

- names a set of attribute instructions that can be called on demand
- uses a namespace-qualified name promoting the easy sharing of stylesheet fragments
- well-defined integration with other methods of specifying attributes for a result element
  - any <xsl:attribute-set> collections named in an xsl:use-attribute-sets= attribute are added first
  - any attribute specifications in the literal result element itself are added next
  - any executed <xsl:attribute> instructions in the literal result element's template are added last
  - earlier defined attribute values are replaced with later defined values without an error
  - last value assigned to an attribute is what remains in the result tree

# Simpler list and footnote structures in XSL-FO

Annex A - Using XSLT with XSL-FO

Section 1 - XSLT language features supporting XSL-FO



The numbering facilities in XSLT allow the list structures in XSL-FO to be more simply defined:

- the collection of list formatting objects are layout oriented, not content oriented
  - a list is a list regardless of how it is labeled or what it contains
- the content of the objects is defined by XSLT, thereby reducing number of objects
  - no distinction between numbered and unnumbered lists
  - the type and structure of the list item labels is entirely out of the scope of the formatting and rendering

The semantics of footnote citation numbering are not part of XSL-FO

- the value of the footnote citation is determined by the transformation process, not by the formatting process
  - a drawback is that this prevents footnote numbering from being page-based

## Common errors writing expressions

Annex A - Using XSLT with XSL-FO

Section 2 - XSL-FO language features similar to XSLT and XPath



---

The expression language is very close, but not identical:

- same operators
- wider set of operands in XSL-FO than XPath

Nuances of differences can make the writing of XSLT confusing

- triggers stylesheet errors

PREVIEW EXCERPT

## Common errors writing expressions (cont.)

Annex A - Using XSLT with XSL-FO

Section 2 - XSL-FO language features similar to XSLT and XPath



Consider the situation of arithmetic calculations with lengths:

```

01 <xsl:template name="test">
02 <xsl:variable name="num" select="100"/>
03 <xsl:variable name="medium-font" select="'10pt'"/>
04
05 <block space-before="10pt div 2">
06 First test: <xsl:value-of select="$num div 2"/>
07 </block>
08
09 <block space-before="{ $medium-font } div 2">
10 Second test
11 </block>
12
13 <block space-before="{ $medium-font div 2 }">
14 Third test
15 </block>
16
17 <block>
18   <xsl:attribute name="space-before">
19     <xsl:value-of select="$medium-font div 2"/>
20   </xsl:attribute>
21   Fourth test
22 </block>
23
24 <block>
25   <xsl:attribute name="space-before">
26     <xsl:value-of select="$medium-font"/> div 2<xsl:text/>
27   </xsl:attribute>
28   Fifth test
29 </block>
30 </xsl:template>

```

There will be no spacing for the third and fourth tests above

- the arithmetic calculation yields the NaN result because a length is not a number in XSLT as in XSL-FO
  - XSLT cannot do arithmetic operations with string operands
- a formatter may not report an error and assume a value of 0 for NaN since it is a valid number in the numbering system
  - no indication to the stylesheet writer that anything is wrong, yet results are not as expected and no error messages to diagnose

## Annex B - XSL-FO expressions



- 
- Introduction - Expressions in XSL-FO
  - Section 1 - XSL-FO expressions
  - Section 2 - XSL-FO functions

PREVIEW EXCERPT

# Expressions in XSL-FO

Annex B - XSL-FO expressions



This annex first summarizes in production order all of the productions in the expression grammar in the Recommendation:

- the left-hand side is the production being defined
- the right-hand side is the definition of the grammatical construct

Secondly, this annex summarizes the built-in functions available in the XSL-FO processor.

PREVIEW EXCERPT

# Production summary

Annex B - XSL-FO expressions  
Section 1 - XSL-FO expressions



[1] Expr	::= AdditiveExpr[11]
[2] PrimaryExpr	::= '(' Expr[1] ')'   Numeric[5]   Literal[20]   Color[18]   Keyword[24]   EnumerationToken[26]   FunctionCall[3]
[3] FunctionCall	::= FunctionName[25] '(' ( Argument[4] ( ',' Argument[4] )*)? ')'
[4] Argument	::= Expr[1]
[5] Numeric	::= AbsoluteNumeric[6]   RelativeNumeric[8]
[6] AbsoluteNumeric	::= AbsoluteLength[7]
[7] AbsoluteLength	::= Number[15] AbsoluteUnitName[27]?
[8] RelativeNumeric	::= Percent[9]   RelativeLength[10]
[9] Percent	::= Number[15] '%'
[10] RelativeLength	::= Number[15] RelativeUnitName[27]
[11] AdditiveExpr	::= MultiplicativeExpr[12]   AdditiveExpr[11] '+' MultiplicativeExpr[12]   AdditiveExpr[11] '-' MultiplicativeExpr[12]
[12] MultiplicativeExpr	::= UnaryExpr[13]   MultiplicativeExpr[12] MultiplyOperator[23] UnaryExpr[13]   MultiplicativeExpr[12] 'div' UnaryExpr[13]   MultiplicativeExpr[12] 'mod' UnaryExpr[13]
[13] UnaryExpr	::= PrimaryExpr[2]   '-' UnaryExpr[13]
[14] ExprToken	::= '('   ')'   '%'   Operator[21]   FunctionName[25]   EnumerationToken[26]   Number[15]
[15] Number	::= FloatingPointNumber[16]
[16] FloatingPointNumber	::= Digits[17] ( '.' Digits[17] )?   '.' Digits[17]
[17] Digits	::= [0-9]+
[18] Color	::= '#' AlphaOrDigits[19]
[19] AlphaOrDigits	::= [a-fA-F0-9]+
[20] Literal	::= "'" [^"]* "'"   '"' [^']* '"'

[21] Operator	::= OperatorName[22]   MultiplyOperator[23]   '+'   '-'
[22] OperatorName	::= 'mod'   'div'
[23] MultiplyOperator	::= '*'
[24] Keyword	::= 'inherit'
[25] FunctionName	::= NCName(XML Namespaces)
[26] EnumerationToken	::= NCName(XML Namespaces)
[27] AbsoluteUnitName	::= 'cm'   'mm'   'in'   'pt'   'pc'   'px'
[28] RelativeUnitName	::= 'em'
[29] ExprWhitespace	::= S(XML)





## Function summary

Annex B - XSL-FO expressions  
Section 2 - XSL-FO functions

Expressions can incorporate numbers and lengths as operands

- a number does not include a unit of measure
  - termed "a unit power of zero"
- a length does include a unit of measure (inches, mm, etc.)
  - termed "a unit power of one"

Performing numeric operations on lengths

- some functions accept either lengths or numbers as operands
  - e.g. `font-size="min(10pt,1em)"`
  - e.g. `<table-cell column-number="min(3,5)">`
  - sometimes (as in `min()`) both operands must have the same unit power
- some functions do not work on lengths, only numbers
  - e.g. `round()` accepts only numbers, not lengths
    - to round a length: convert length to a number, round, then convert result to a length
  - e.g. `round(.75in div 1in)*1in`

Percentages are counted in 1/100 units

- can be utilized in a property as a relation to the property's current value
  - the following expressions all evaluate to the same value:
    - `font-size="150%"`
    - `font-size="1.5 * inherited-property-value(font-size)"`
    - `font-size="1.5em"`

# Function groupings

Annex B - XSL-FO expressions  
Section 2 - XSL-FO functions



## Color Functions (5.10.2)

- `rgb()`
- `rgb-icc()`
- `system-color()`

## Font Functions (5.10.3)

- `system-font()`

## Number Functions (5.10.1)

- `abs()`
- `ceiling()`
- `floor()`
- `max()`
- `min()`
- `round()`

## Property Value Functions (5.10.4)

- `body-start()`
- `from-nearest-specified-value()`
- `from-page-master-region()`
- `from-parent()`
- `from-table-column()`
- `inherited-property-value()`
- `label-end()`
- `merge-property-values()`
- `proportional-column-width()`

# Functions summarized by name

Annex B - XSL-FO expressions  
Section 2 - XSL-FO functions



PREVIEW EXCERPT

numeric abs(numeric) (5.10.1)

- returns absolute value of number or length argument

numeric body-start() (5.10.4)

- returns the start indent corresponding to the list-item body when considering the provisional-valued properties of the list

numeric ceiling(numeric) (5.10.1)

- returns number closest to positive infinity

numeric floor(numeric) (5.10.1)

- returns number closest to negative infinity

object from-nearest-specified-value(NCName) (5.10.4)

- returns the value of the named property from the nearest ancestor of the formatting object in which the property value is being explicitly specified
- returns the initial value if there is no parent

object from-page-master-region(NCName) (5.10.4)

- signals for each region created for a page sequence that the property's value is obtained from the named property of the region's definition, or if no name is supplied, the property being assigned
- until used differently in some future specification, XSL-FO 1.1 restricts this only to the two properties `writeing-mode=` and `reference-orientation=`, and in both cases a named property is not allowed to be specified

object from-parent(NCName) (5.10.4)

- returns the value of the named property from the parent of the formatting object of the property being evaluated
- returns the initial value if there is no parent

object from-table-column(NCName) (5.10.4)

- returns the value of the named property from the `<table-column>` corresponding to the current column in the table

object inherited-property-value(NCName) (5.10.4)

- returns the value of the inherited property named
- it is an error if the property named is not an inherited property

numeric label-end() (5.10.4)

- returns the end indent corresponding to the list-item label when considering the provisional-valued properties of the list

numeric max(numeric, numeric) (5.10.1)

- returns the minimum of two arguments
- arguments may be either lengths or numbers but must be the same type

object merge-property-values(NCName) (5.10.4)

- returns the property calculated from the property set corresponding to the current user-agent state within the parent's child property sets

numeric min(numeric, numeric) (5.10.1)

- returns the minimum of two arguments
- arguments may be either lengths or numbers but must be the same type

numeric proportional-column-width(numeric) (5.10.4)

- returns the length corresponding to the number of units of proportional measure of the current table column's table as indicated in the supplied argument
- note that proportional measure is that length left over when removing specified column widths from the table width and dividing the result by the number of columns for which widths are not specified

color rgb(numeric, numeric, numeric) (5.10.2)

- returns a color from the RGB space
- arguments must be numbers, not lengths

color rgb-icc(numeric, numeric, numeric, NCName, numeric, numeric) (5.10.2)

- returns a color from the named ICC color profile
- the first three arguments are the fallback RGB color if the ICC color is not found
- the last arguments are specific to the color profile

numeric round(numeric) (5.10.1)

- returns whole number closest to given number
- returns ceiling() on the value .5

color system-color(NCName) (5.10.2)

- returns the system-defined color named in the argument

object system-font(NCName, NCName) (5.10.3)

- returns the font-size characteristic named in the second argument of the system font named in the first argument
- if the second argument is omitted, the characteristic is that which is being assigned by the expression

## Annex C - XSL-FO object summary



- 
- Introduction - Objects
  - Section 1 - Formatting objects

PREVIEW EXCERPT

# Objects

Annex C - XSL-FO object summary



---

This annex first summarizes in alphabetical order all of the formatting objects in the Recommendation:

- the citation on the first line is to the section of the W3C Recommendation document
- the citation on the second line is to the section of this material

Secondly, this annex summarizes the objects by their type as categorized by the Recommendation:

- the citation on each subordinate line is to the section of this material

Lastly, a prototypical arrangement of all formatting objects is shown to indicate the nesting of child objects inside of expected or typical parent objects.

# Objects summarized by name

Annex C - XSL-FO object summary  
Section 1 - Formatting objects




---

<basic-link>(6.9.2;87) (visual: extended; aural: extended)  
- (#PCDATA|%inline;|%block;)\*

<bidirectional-override>(6.6.2;119) (visual: extended; aural: basic)  
- (#PCDATA|%inline;|%block;)\*

<block>(6.5.2;81) (visual: basic; aural: basic)  
- (#PCDATA|%inline;|%block;)\*

<block-container>(6.5.3;98) (visual: extended; aural: basic)  
- (%block;)+

¶ <bookmark>(6.11.2;113) (visual: extended; aural: extended)  
- (bookmark-title,bookmark\*)

¶ <bookmark-title>(6.11.3;113) (visual: extended; aural: extended)  
- (#PCDATA)

¶ <bookmark-tree>(6.11.1;113) (visual: extended; aural: extended)  
- (bookmark+)

¶ <change-bar-begin>(6.13.2;119) (visual: extended; aural: extended)  
- EMPTY

¶ <change-bar-end>(6.13.3;119) (visual: extended; aural: extended)  
- EMPTY

<character>(6.6.3;119) (visual: basic; aural: basic)  
- EMPTY

<color-profile>(6.4.4;119) (visual: extended; aural: N/A)  
- EMPTY

<conditional-page-master-reference>(6.4.12;105) (visual: extended; aural: extended)  
- EMPTY

<declarations>(6.4.3;119) (visual: basic; aural: basic)  
- (color-profile)\*

<external-graphic>(6.6.5;87) (visual: basic; aural: basic)  
- EMPTY

<float>(6.12.2;98) (visual: extended; aural: extended)  
- (%block;)+

<flow>(6.4.19;68) (visual: basic; aural: basic)  
- (%block;)+

¶ <flow-assignment>(6.4.23;105) (visual: extended; aural: extended)  
- (flow-source-list,flow-target-list)



¶ <flow-map>(6.4.22;105) (visual: extended; aural: extended)  
     - (flow-assignment+)  
 ¶ <flow-name-specifier>(6.4.25;105) (visual: extended; aural: extended)  
     - EMPTY  
 ¶ <flow-source-list>(6.4.24;105) (visual: extended; aural: extended)  
     - (flow-name-specifier+)  
 ¶ <flow-target-list>(6.4.26;105) (visual: extended; aural: extended)  
     - (region-name-specifier+)  
 ¶ <folio-prefix>(6.6.13;104) (visual: extended; aural: extended)  
     - (#PCDATA|%inline;)\*  
 ¶ <folio-suffix>(6.6.14;104) (visual: extended; aural: extended)  
     - (#PCDATA|%inline;)\*  
 <footnote>(6.12.3;98) (visual: extended; aural: extended)  
     - (inline,footnote-body)  
 <footnote-body>(6.12.4;98) (visual: extended; aural: extended)  
     - (%block;)+  
 ¶ <index-key-reference>(6.10.6;113) (visual: extended; aural: extended)  
     - (index-page-number-prefix?,index-page-number-suffix?)  
 ¶ <index-page-citation-list>(6.10.7;113) (visual: extended; aural: extended)  
     - (index-page-citation-list-separator?,  
       index-page-citation-range-separator?,index-key-reference+)  
 ¶ <index-page-citation-list-separator>(6.10.8;113) (visual: extended; aural: extended)  
     - (#PCDATA|%inline;)\*  
 ¶ <index-page-citation-range-separator>(6.10.9;113) (visual: extended; aural: extended)  
     - (#PCDATA|%inline;)\*  
 ¶ <index-page-number-prefix>(6.10.2;113) (visual: extended; aural: extended)  
     - (#PCDATA|%inline;)\*  
 ¶ <index-page-number-suffix>(6.10.3;113) (visual: extended; aural: extended)  
     - (#PCDATA|%inline;)\*  
 ¶ <index-range-begin>(6.10.4;113) (visual: extended; aural: extended)  
     - EMPTY  
 ¶ <index-range-end>(6.10.5;113) (visual: extended; aural: extended)  
     - EMPTY  
 <initial-property-set>(6.6.4;81) (visual: extended; aural: basic)  
     - EMPTY  
 <inline>(6.6.7;81) (visual: basic; aural: basic)  
     - (#PCDATA|%inline;|%block;)\*

<inline-container>(6.6.8;98) (visual: extended; aural: extended)  
 - (%block;)+

<instream-foreign-object>(6.6.6;87) (visual: extended; aural: extended)  
 - a single child element from a non-XSL namespace

<layout-master-set>(6.4.7;63) (visual: basic; aural: basic)  
 - (simple-page-master|page-sequence-master|flow-map)+

<leader>(6.6.9;87) (visual: basic; aural: basic)  
 - (#PCDATA|%inline;)\*

<list-block>(6.8.2;87) (visual: basic; aural: basic)  
 - (list-item+)

<list-item>(6.8.3;87) (visual: basic; aural: basic)  
 - (list-item-label,list-item-body)

<list-item-body>(6.8.4;87) (visual: basic; aural: basic)  
 - (%block;)+

<list-item-label>(6.8.5;87) (visual: extended; aural: basic)  
 - (%block;)+

<marker>(6.13.5;104) (visual: extended; aural: extended)  
 - (#PCDATA|%inline;|%block;)\*

<multi-case>(6.9.4;122) (visual: basic; aural: basic)  
 - (#PCDATA|%inline;|%block;)\*

<multi-properties>(6.9.6;122) (visual: extended, need not be implemented for extended conformance for non-interactive media; aural: extended)  
 - (multi-property-set+,wrapper)

<multi-property-set>(6.9.7;122) (visual: extended, need not be implemented for extended conformance for non-interactive media; aural: extended)  
 - EMPTY

<multi-switch>(6.9.3;122) (visual: extended, need not be implemented for extended conformance for non-interactive media; aural: extended)  
 - (multi-case+)

<multi-toggle>(6.9.5;122) (visual: extended, need not be implemented for extended conformance for non-interactive media; aural: extended)  
 - (#PCDATA|%inline;|%block;)\*

<page-number>(6.6.10;104) (visual: basic; aural: extended)  
 - EMPTY

<page-number-citation>(6.6.11;81) (visual: extended; aural: extended)  
 - EMPTY

¶ <page-number-citation-last>(6.6.12;81) (visual: extended; aural: extended)  
 - EMPTY

<page-sequence>(6.4.5;65) (visual: basic; aural: basic)  
 - (title?,folio-prefix?,folio-suffix?,static-content\*,flow+)

<page-sequence-master>(6.4.8;105) (visual: basic; aural: basic)  
 - (single-page-master-reference|repeatable-page-master-reference|repeatable-page-master-alternatives)+

❶ <page-sequence-wrapper>(6.4.6;67) (visual: basic; aural: basic)  
 - (page-sequence|page-sequence-wrapper)\*

<region-after>(6.4.16;104) (visual: extended; aural: extended)  
 - EMPTY

<region-before>(6.4.15;104) (visual: extended; aural: extended)  
 - EMPTY

<region-body>(6.4.14;81) (visual: basic; aural: basic)  
 - EMPTY

<region-end>(6.4.18;104) (visual: extended; aural: extended)  
 - EMPTY

❶ <region-name-specifier>(6.4.27;105) (visual: extended; aural: extended)  
 - EMPTY

<region-start>(6.4.17;104) (visual: extended; aural: extended)  
 - EMPTY

<repeatable-page-master-alternatives>(6.4.11;105) (visual: extended; aural: extended)  
 - (conditional-page-master-reference+)

<repeatable-page-master-reference>(6.4.10;105) (visual: basic; aural: basic)  
 - EMPTY

<retrieve-marker>(6.13.6;104) (visual: extended; aural: extended)  
 - EMPTY

❶ <retrieve-table-marker>(6.13.7;104) (visual: extended; aural: extended)  
 - EMPTY

<root>(6.4.2;61) (visual: basic; aural: basic)  
 - (layout-master-set,declarations?,bookmark-tree?,(page-sequence|page-sequence-wrapper)+)

❶ <scaling-value-citation>(6.6.15;87) (visual: extended; aural: extended)  
 - EMPTY

<simple-page-master>(6.4.13;81) (visual: basic; aural: basic)  
 - (region-body+,region-before?,region-after?,region-start?,region-end?)

<single-page-master-reference>(6.4.9;105) (visual: basic; aural: basic)  
 - EMPTY

<static-content>(6.4.20;104) (visual: extended; aural: extended)  
 - (%block;)+

<table>(6.7.3;93) (visual: basic; aural: basic)  
 - (table-column\*,table-header?,table-footer?,table-body+)  
 <table-and-caption>(6.7.2;93) (visual: basic; aural: basic)  
 - (table-caption?,table)  
 <table-body>(6.7.8;93) (visual: basic; aural: basic)  
 - (table-row+|table-cell+)  
 <table-caption>(6.7.5;93) (visual: extended; aural: extended)  
 - (%block;)+  
 <table-cell>(6.7.10;93) (visual: basic; aural: basic)  
 - (%block;)+  
 <table-column>(6.7.4;93) (visual: basic; aural: basic)  
 - EMPTY  
 <table-footer>(6.7.7;93) (visual: extended; aural: extended)  
 - (table-row+|table-cell+)  
 <table-header>(6.7.6;93) (visual: basic; aural: basic)  
 - (table-row+|table-cell+)  
 <table-row>(6.7.9;93) (visual: basic; aural: basic)  
 - (table-cell+)  
 <title>(6.4.21;81) (visual: extended; aural: extended)  
 - (#PCDATA|%inline;)\*  
 <wrapper>(6.13.4;81) (visual: basic; aural: basic)  
 - (#PCDATA|%inline;|%block;)\*

## Prototypical hierarchy

Annex C - XSL-FO object summary  
Section 1 - Formatting objects



All formatting objects are listed below in a typical nesting of parents and children. Where sibling children order is not significant, an ellipsis is shown between them. Not all nuances

PREVIEW EXCERPT

of nesting are reflected in this hierarchy as this is to be regarded more as a guideline than as a specification.

PREVIEW EXCERPT

- <root>(6.4.2;61)
  - <layout-master-set>(6.4.7;63)
    - <simple-page-master>(6.4.13;81)
      - <region-body />(6.4.14;81)
      - <region-before />(6.4.15;104)
      - <region-after />(6.4.16;104)
      - <region-start />(6.4.17;104)
      - <region-end />(6.4.18;104)
    - ...
    - <page-sequence-master>(6.4.8;105)
      - <single-page-master-reference />(6.4.9;105)
      - ...
      - <repeatable-page-master-reference />(6.4.10;105)
      - ...
      - <repeatable-page-master-alternatives>(6.4.11;105)
        - <conditional-page-master-reference />(6.4.12;105)
    - ...
    - ① <flow-map>(6.4.22;105)
      - ① <flow-assignment>(6.4.23;105)
        - ① <flow-source-list>(6.4.24;105)
          - ① <flow-name-specifier />(6.4.25;105)
        - ① <flow-target-list>(6.4.26;105)
          - ① <region-name-specifier />(6.4.27;105)
  - <declarations>(6.4.3;119)
    - <color-profile />(6.4.4;119)
  - ① <bookmark-tree>(6.11.1;113)
    - ① <bookmark>(6.11.2;113)
      - ① <bookmark-title>(6.11.3;113)
      - ① <bookmark>(6.11.2;113)
      - ...
  - ① <page-sequence-wrapper>(6.4.6;67)
    - <page-sequence>(6.4.5;65)
    - ① <page-sequence-wrapper>(6.4.6;67)
    - ...
  - <page-sequence>(6.4.5;65)
    - <title>(6.4.21;81)
    - ① <folio-prefix>(6.6.13;104)
      - %inline;
    - ① <folio-suffix>(6.6.14;104)
      - %inline;
    - <static-content>(6.4.20;104)
      - <retrieve-marker />(6.13.6;104)
      - ...
      - <page-number />(6.6.10;104)
    - <flow>(6.4.19;68)
      - <block-container>(6.5.3;98)

- %block;
- ...
- <float>(6.12.2;98)
  - %block;
- ...
- <table-and-caption>(6.7.2;93)
  - <table-caption>(6.7.5;93)
  - <table>(6.7.3;93)
    - <table-column/>(6.7.4;93)
    - <table-header>(6.7.6;93)
      - ① <retrieve-table-marker/>(6.13.7;104)
    - <table-footer>(6.7.7;93)
      - ① <retrieve-table-marker/>(6.13.7;104)
    - <table-body>(6.7.8;93)
      - <table-row>(6.7.9;93)
      - <table-cell>(6.7.10;93)
        - %block;
- ...
- <list-block>(6.8.2;87)
  - <list-item>(6.8.3;87)
    - <list-item-label>(6.8.5;87)
      - %block;
    - <list-item-body>(6.8.4;87)
      - %block;
- ...
- <wrapper>(6.13.4;81)
  - %block;
- ...



- <block>(6.5.2;81)
  - <marker>(6.13.5;104)
  - <initial-property-set />(6.6.4;81)
  - <inline-container>(6.6.8;98)
    - %block;
  - ...
  - <basic-link>(6.9.2;87)
  - ...
  - <bidirectional-override>(6.6.2;119)
  - ...
  - ¶ <change-bar-begin />(6.13.2;119)
  - ...
  - ¶ <change-bar-end />(6.13.3;119)
  - ...
  - <character />(6.6.3;119)
  - ...
  - <external-graphic />(6.6.5;87)
  - ...
  - <footnote>(6.12.3;98)
    - <footnote-body>(6.12.4;98)
      - %block;
  - ...
  - ¶ <index-page-citation-list>(6.10.7;113)
    - ¶ <index-page-citation-list-separator>(6.10.8;113)
      - %inline;
    - ¶ <index-page-citation-range-separator>(6.10.9;113)
      - %inline;
    - ¶ <index-key-reference>(6.10.6;113)
      - ¶ <index-page-number-prefix>(6.10.2;113)
        - %inline;
      - ¶ <index-page-number-suffix>(6.10.3;113)
        - %inline;
    - ...
    - ¶ <index-range-begin />(6.10.4;113)
    - ...
    - ¶ <index-range-end />(6.10.5;113)
    - ...
    - <inline>(6.6.7;81)
    - ...
    - <instream-foreign-object>(6.6.6;87)
    - ...
    - <leader>(6.6.9;87)
    - ...
    - <page-number-citation />(6.6.11;81)
    - ...
    - ¶ <page-number-citation-last />(6.6.12;81)

- ...
- ① <scaling-value-citation/>(6.6.15;87)
- ...
- <wrapper>(6.13.4;81)
  - %inline;
- ...
- <multi-switch>(6.9.3;122)
  - <multi-case>(6.9.4;122)
  - <multi-toggle>(6.9.5;122)
- ...
- <multi-properties>(6.9.6;122)
  - <multi-property-set/>(6.9.7;122)

# Objects summarized by type

Annex C - XSL-FO object summary  
Section 1 - Formatting objects



## Block-level Formatting Objects(6.5)

- <block>(6.5.2;81)
- <block-container>(6.5.3;98)

## Declarations and Pagination and Layout Formatting Objects(6.4)

- <color-profile>(6.4.4;119)
- <conditional-page-master-reference>(6.4.12;105)
- <declarations>(6.4.3;119)
- <flow>(6.4.19;68)
- ¶ <flow-assignment>(6.4.23;105)
- ¶ <flow-map>(6.4.22;105)
- ¶ <flow-name-specifier>(6.4.25;105)
- ¶ <flow-source-list>(6.4.24;105)
- ¶ <flow-target-list>(6.4.26;105)
- <layout-master-set>(6.4.7;63)
- <page-sequence>(6.4.5;65)
- <page-sequence-master>(6.4.8;105)
- ¶ <page-sequence-wrapper>(6.4.6;67)
- <region-after>(6.4.16;104)
- <region-before>(6.4.15;104)
- <region-body>(6.4.14;81)
- <region-end>(6.4.18;104)
- ¶ <region-name-specifier>(6.4.27;105)
- <region-start>(6.4.17;104)
- <repeatable-page-master-alternatives>(6.4.11;105)
- <repeatable-page-master-reference>(6.4.10;105)
- <root>(6.4.2;61)
- <simple-page-master>(6.4.13;81)
- <single-page-master-reference>(6.4.9;105)
- <static-content>(6.4.20;104)
- <title>(6.4.21;81)

## Dynamic Effects: Link and Multi Formatting Objects(6.9)

- <basic-link>(6.9.2;87)
- <multi-case>(6.9.4;122)
- <multi-properties>(6.9.6;122)
- <multi-property-set>(6.9.7;122)
- <multi-switch>(6.9.3;122)
- <multi-toggle>(6.9.5;122)

## Formatting Objects for Bookmarks(6.11)

- `<bookmark>`(6.11.2;113)
- `<bookmark-title>`(6.11.3;113)
- `<bookmark-tree>`(6.11.1;113)

## Formatting Objects for Indexing(6.10)

- `<index-key-reference>`(6.10.6;113)
- `<index-page-citation-list>`(6.10.7;113)
- `<index-page-citation-list-separator>`(6.10.8;113)
- `<index-page-citation-range-separator>`(6.10.9;113)
- `<index-page-number-prefix>`(6.10.2;113)
- `<index-page-number-suffix>`(6.10.3;113)
- `<index-range-begin>`(6.10.4;113)
- `<index-range-end>`(6.10.5;113)

## Formatting Objects for Lists(6.8)

- `<list-block>`(6.8.2;87)
- `<list-item>`(6.8.3;87)
- `<list-item-body>`(6.8.4;87)
- `<list-item-label>`(6.8.5;87)

## Formatting Objects for Tables(6.7)

- `<table>`(6.7.3;93)
- `<table-and-caption>`(6.7.2;93)
- `<table-body>`(6.7.8;93)
- `<table-caption>`(6.7.5;93)
- `<table-cell>`(6.7.10;93)
- `<table-column>`(6.7.4;93)
- `<table-footer>`(6.7.7;93)
- `<table-header>`(6.7.6;93)
- `<table-row>`(6.7.9;93)

## Inline-level Formatting Objects(6.6)

- `<bidirectional-override>`(6.6.2;119)
- `<character>`(6.6.3;119)
- `<external-graphic>`(6.6.5;87)
- `<folio-prefix>`(6.6.13;104)
- `<folio-suffix>`(6.6.14;104)
- `<initial-property-set>`(6.6.4;81)
- `<inline>`(6.6.7;81)
- `<inline-container>`(6.6.8;98)
- `<instream-foreign-object>`(6.6.6;87)
- `<leader>`(6.6.9;87)
- `<page-number>`(6.6.10;104)
- `<page-number-citation>`(6.6.11;81)
- `<page-number-citation-last>`(6.6.12;81)
- `<scaling-value-citation>`(6.6.15;87)

## Other Formatting Objects(6.13)

- ¶ <change-bar-begin>(6.13.2;119)
- ¶ <change-bar-end>(6.13.3;119)
- <marker>(6.13.5;104)
- <retrieve-marker>(6.13.6;104)
- ¶ <retrieve-table-marker>(6.13.7;104)
- <wrapper>(6.13.4;81)

## Out-of-Line Formatting Objects(6.12)

- <float>(6.12.2;98)
- <footnote>(6.12.3;98)
- <footnote-body>(6.12.4;98)

## Annex D - XSL-FO property summaries



- 
- Introduction - Properties
  - Section 1 - Property groupings
  - Section 2 - Data types
  - Section 3 - Property summaries

PREVIEW EXCERPT

# Properties

Annex D - XSL-FO property summaries



---

This annex first summarizes the collections of common properties.

This annex then summarizes all the data types of the properties.

This annex then summarizes all shorthand and individual properties each in alphabetical order:

- the citation is to the W3C Recommendation
- after the citation is either "CSS" or "XSL" indicating the heritage of this property

Note that much of this annex is synthesized from the application of XSLT to the source XML of the W3C Recommendation document itself, demonstrating the power of using these technologies for information description.

PREVIEW EXCERPT

# Common Absolute Position Properties

Annex D - XSL-FO property summaries  
Section 1 - Property groupings



## Properties (7.6):

`absolute-position=(7.6.1;181)`

`bottom=(7.6.4;196)`

`left=(7.6.5;213)`

`right=(7.6.3;227)`

`top=(7.6.2;236)`

## Referring object:

`<block-container>(6.5.3;98)`

## Shorthand influencing the above properties:

`position=(7.31.20;224)`



# Common Accessibility Properties

Annex D - XSL-FO property summaries  
Section 1 - Property groupings



## Properties (7.5):

role=(7.5.2;228)

source-document=(7.5.1;229)

## Referring objects:

<basic-link>(6.9.2;87)

<block>(6.5.2;81)

❶ <bookmark>(6.11.2;113)

❶ <bookmark-title>(6.11.3;113)

❶ <change-bar-begin>(6.13.2;119)

❶ <change-bar-end>(6.13.3;119)

<external-graphic>(6.6.5;87)

<footnote>(6.12.3;98)

<footnote-body>(6.12.4;98)

<initial-property-set>(6.6.4;81)

<inline>(6.6.7;81)

<instream-foreign-object>(6.6.6;87)

<leader>(6.6.9;87)

<list-block>(6.8.2;87)

<list-item>(6.8.3;87)

<list-item-body>(6.8.4;87)

<list-item-label>(6.8.5;87)

<multi-case>(6.9.4;122)

<multi-properties>(6.9.6;122)

<multi-switch>(6.9.3;122)

<multi-toggle>(6.9.5;122)

<page-number>(6.6.10;104)

<page-number-citation>(6.6.11;81)

❶ <page-number-citation-last>(6.6.12;81)

<root>(6.4.2;61)

❶ <scaling-value-citation>(6.6.15;87)

<table>(6.7.3;93)

<table-and-caption>(6.7.2;93)

<table-body>(6.7.8;93)

<table-caption>(6.7.5;93)

<table-cell>(6.7.10;93)

<table-footer>(6.7.7;93)

<table-header>(6.7.6;93)

<table-row>(6.7.9;93)

<title>(6.4.21;81)

# Common Aural Properties

Annex D - XSL-FO property summaries  
Section 1 - Property groupings



## Properties (7.7):

azimuth=(7.7.1;183)  
cue-after=(7.7.2;200)  
cue-before=(7.7.3;201)  
elevation=(7.7.4;202)  
pause-after=(7.7.5;222)  
pause-before=(7.7.6;223)  
pitch=(7.7.7;223)  
pitch-range=(7.7.8;223)  
play-during=(7.7.9;224)

richness=(7.7.10;227)  
speak=(7.7.11;231)  
speak-header=(7.7.12;231)  
speak-numeral=(7.7.13;231)  
speak-punctuation=(7.7.14;231)  
speech-rate=(7.7.15;232)  
stress=(7.7.16;233)  
voice-family=(7.7.17;237)  
volume=(7.7.18;237)

## Referring objects:

<basic-link>(6.9.2;87)  
<bidirectional-override>(6.6.2;119)  
<block>(6.5.2;81)  
❶ <change-bar-begin>(6.13.2;119)  
❶ <change-bar-end>(6.13.3;119)  
<character>(6.6.3;119)  
<external-graphic>(6.6.5;87)  
<initial-property-set>(6.6.4;81)  
<inline>(6.6.7;81)  
<instream-foreign-object>(6.6.6;87)  
<leader>(6.6.9;87)  
<list-block>(6.8.2;87)  
<list-item>(6.8.3;87)

<page-number>(6.6.10;104)  
<page-number-citation>(6.6.11;81)  
❶ <page-number-citation-last>(6.6.12;81)  
❶ <scaling-value-citation>(6.6.15;87)  
<table>(6.7.3;93)  
<table-and-caption>(6.7.2;93)  
<table-body>(6.7.8;93)  
<table-caption>(6.7.5;93)  
<table-cell>(6.7.10;93)  
<table-footer>(6.7.7;93)  
<table-header>(6.7.6;93)  
<table-row>(6.7.9;93)  
<title>(6.4.21;81)

## Shorthands influencing the above properties:

cue=(7.31.12;200)      pause=(7.31.19;222)



# Common Border, Padding, and Background Properties

Annex D - XSL-FO property summaries  
Section 1 - Property groupings

## Properties (7.8):

background-attachment=(7.8.1;183)	border-left-style=(7.8.26;192)
background-color=(7.8.2;184)	border-left-width=(7.8.27;192)
background-image=(7.8.3;184)	border-right-color=(7.8.28;192)
background-position-horizontal=(7.8.5;185)	border-right-style=(7.8.29;193)
background-position-vertical=(7.8.6;185)	border-right-width=(7.8.30;193)
background-repeat=(7.8.4;185)	border-start-color=(7.8.13;194)
border-after-color=(7.8.10;186)	border-start-style=(7.8.14;194)
border-after-style=(7.8.11;187)	border-start-width=(7.8.15;194)
border-after-width=(7.8.12;187)	border-top-color=(7.8.19;195)
border-before-color=(7.8.7;188)	border-top-style=(7.8.20;195)
border-before-style=(7.8.8;188)	border-top-width=(7.8.21;196)
border-before-width=(7.8.9;188)	padding-after=(7.8.32;218)
border-bottom-color=(7.8.22;189)	padding-before=(7.8.31;219)
border-bottom-style=(7.8.23;189)	padding-bottom=(7.8.36;219)
border-bottom-width=(7.8.24;189)	padding-end=(7.8.34;219)
border-end-color=(7.8.16;190)	padding-left=(7.8.37;220)
border-end-style=(7.8.17;191)	padding-right=(7.8.38;220)
border-end-width=(7.8.18;191)	padding-start=(7.8.33;220)
border-left-color=(7.8.25;191)	padding-top=(7.8.35;221)

## Referring objects:

<basic-link>(6.9.2;87)	<region-before>(6.4.15;104)
<block>(6.5.2;81)	<region-body>(6.4.14;81)
<block-container>(6.5.3;98)	<region-end>(6.4.18;104)
<character>(6.6.3;119)	<region-start>(6.4.17;104)
<external-graphic>(6.6.5;87)	❶ <scaling-value-citation>(6.6.15;87)
<initial-property-set>(6.6.4;81)	<table>(6.7.3;93)
<inline>(6.6.7;81)	<table-and-caption>(6.7.2;93)
<inline-container>(6.6.8;98)	<table-body>(6.7.8;93)
<instream-foreign-object>(6.6.6;87)	<table-caption>(6.7.5;93)
<leader>(6.6.9;87)	<table-cell>(6.7.10;93)
<list-block>(6.8.2;87)	<table-column>(6.7.4;93)
<list-item>(6.8.3;87)	<table-footer>(6.7.7;93)
<page-number>(6.6.10;104)	<table-header>(6.7.6;93)
<page-number-citation>(6.6.11;81)	<table-row>(6.7.9;93)
❶ <page-number-citation-last>(6.6.12;81)	<title>(6.4.21;81)
<region-after>(6.4.16;104)	

## Shorthands influencing the above properties:

background=(7.31.1;183)	border-right=(7.31.7;192)
background-position=(7.31.2;184)	border-style=(7.31.8;195)
border=(7.31.3;186)	border-top=(7.31.10;195)

border-bottom=(7.31.4;189)  
border-color=(7.31.5;190)  
border-left=(7.31.6;191)

border-width=(7.31.11;196)  
padding=(7.31.15;218)

PREVIEW EXCERPT

# Common Font Properties

Annex D - XSL-FO property summaries  
Section 1 - Property groupings



## Properties (7.9):

font-family=(7.9.2;204)  
font-selection-strategy=(7.9.3;205)  
font-size=(7.9.4;205)  
font-size-adjust=(7.9.6;205)

font-stretch=(7.9.5;206)  
font-style=(7.9.7;206)  
font-variant=(7.9.8;206)  
font-weight=(7.9.9;206)

## Referring objects:

<bidirectional-override>(6.6.2;119)  
<block>(6.5.2;81)  
<character>(6.6.3;119)  
<initial-property-set>(6.6.4;81)  
<inline>(6.6.7;81)  
<leader>(6.6.9;87)

<page-number>(6.6.10;104)  
<page-number-citation>(6.6.11;81)  
① <page-number-citation-last>(6.6.12;81)  
① <scaling-value-citation>(6.6.15;87)  
<title>(6.4.21;81)

## Shorthand influencing the above properties:

font=(7.31.13;204)

# Common Hyphenation Properties

Annex D - XSL-FO property summaries  
Section 1 - Property groupings



## Properties (7.10):

country=(7.10.1;200)

hyphenate=(7.10.4;208)

hyphenation-character=(7.10.5;208)

hyphenation-push-character-count=(7.10.6;208)

hyphenation-remain-character-count=(7.10.7;209)

language=(7.10.2;212)

script=(7.10.3;229)

## Referring objects:

<block>(6.5.2;81)

<character>(6.6.3;119)

## Shorthand influencing the above properties:

❶ xml:lang=(7.31.24;239)

# Common Margin Properties-Block

Annex D - XSL-FO property summaries  
Section 1 - Property groupings



## Properties (7.11):

end-indent=(7.11.8;203)  
margin-bottom=(7.11.2;214)  
margin-left=(7.11.3;215)  
margin-right=(7.11.4;215)

margin-top=(7.11.1;215)  
space-after=(7.11.6;230)  
space-before=(7.11.5;230)  
start-indent=(7.11.7;232)

## Referring objects:

<block>(6.5.2;81)  
<block-container>(6.5.3;98)  
<list-block>(6.8.2;87)  
<list-item>(6.8.3;87)

<region-body>(6.4.14;81)  
<simple-page-master>(6.4.13;81)  
<table>(6.7.3;93)  
<table-and-caption>(6.7.2;93)

## Shorthand influencing the above properties:

margin=(7.31.14;214)

# Common Margin Properties-Inline

Annex D - XSL-FO property summaries  
Section 1 - Property groupings



## Properties (7.12):

space-end=(7.12.5;230)

space-start=(7.12.6;230)

## Referring objects:

<basic-link>(6.9.2;87)

<leader>(6.6.9;87)

<character>(6.6.3;119)

<page-number>(6.6.10;104)

<external-graphic>(6.6.5;87)

<page-number-citation>(6.6.11;81)

<inline>(6.6.7;81)

¶ <page-number-citation-last>(6.6.12;81)

<inline-container>(6.6.8;98)

¶ <scaling-value-citation>(6.6.15;87)

<instream-foreign-object>(6.6.6;87)

<title>(6.4.21;81)



# Common Relative Position Properties

Annex D - XSL-FO property summaries  
Section 1 - Property groupings



## Property (7.13):

relative-position=(7.13.5;226)

## Referring objects:

<basic-link>(6.9.2;87)	<page-number>(6.6.10;104)
<bidirectional-override>(6.6.2;119)	<page-number-citation>(6.6.11;81)
<block>(6.5.2;81)	¶ <page-number-citation-last>(6.6.12;81)
<character>(6.6.3;119)	¶ <scaling-value-citation>(6.6.15;87)
<external-graphic>(6.6.5;87)	<table>(6.7.3;93)
<initial-property-set>(6.6.4;81)	<table-and-caption>(6.7.2;93)
<inline>(6.6.7;81)	<table-body>(6.7.8;93)
<inline-container>(6.6.8;98)	<table-caption>(6.7.5;93)
<instream-foreign-object>(6.6.6;87)	<table-cell>(6.7.10;93)
<leader>(6.6.9;87)	<table-footer>(6.7.7;93)
<list-block>(6.8.2;87)	<table-header>(6.7.6;93)
<list-item>(6.8.3;87)	<table-row>(6.7.9;93)

## Shorthand influencing the above property:

position=(7.31.20;224)

## Property data types

Annex D - XSL-FO property summaries  
Section 2 - Data types



The typographical convention in the XSL-FO 1.0 Recommendation for data types is to use angle brackets around the data type name using a proportional font. This is not to be confused with XSL-FO elements that are also documented using angle brackets, but elements use a monospaced font. The Recommendation convention for this is used here for consistency for those readers referring to the W3C documentation.

PREVIEW EXCERPT

A listing of data types:

**<angle>**

- A representation of an angle consisting of an optional '+' or '-' character immediately followed by a <number> immediately followed by an angle unit identifier. Angle unit identifiers are: 'deg' (for degrees), 'grad' (for grads), and 'rad' (for radians). The specified values are normalized to the range 0deg to 360deg. A property may define additional constraints on the value.

**<character>**

- A single Unicode character valid in accordance with production [2] of XML or XML 1.1. For example, "c" or "&#x2202;".

**<color>**

- Either a string of characters representing a keyword or a color function defined in Color Functions. The list of keyword color names is: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.

**<country>**

- A string of characters conforming to an ISO 3166 (ISO3166-1, ISO3166-2, and ISO3166-3) country code.

**<family-name>**

- A string of characters identifying a font.

**<frequency>**

- A <number> immediately followed by a frequency unit identifier. Frequency unit identifiers are: 'Hz' (for Hertz) and 'kHz' (for kilo Hertz).

**<id>**

- A string of characters conforming to the definition of an NCName in XML Names or XML Names 1.1 and is unique within the stylesheet.

**<idref>**

- A string of characters conforming to the definition of an NCName in XML Names or XML Names 1.1 and that matches an ID property value used within the stylesheet.

**<integer>**

- A signed integer value which consists of an optional '+' or '-' character followed by a sequence of digits. A property may define additional constraints on the value.

**<keep>**

- A compound datatype, with components: within-line, within-column, and within-page. The value of each component is either "auto", "always", or an <integer>.

**<language>**

- A string of characters conforming to either a ISO639-2 3-letter terminology or bibliographic code or a ISO639 2-letter code representing the name of a language.

**<length-bp-ip-direction>**

- A compound datatype, with components: block-progression-direction, and inline-progression-direction. Each component is a `<length>`. A property may define additional constraints on the values.

`<length-conditional>`

- A compound datatype, with components: length, conditionality. The length component is a `<length>`. The conditionality component is either "discard" or "retain". A property may define additional constraints on the values.

`<length-range>`

- A compound datatype, with components: minimum, optimum, maximum. Each component is a `<length>`. If "minimum" is greater than optimum, it will be treated as if it had been set to "optimum". If "maximum" is less than optimum, it will be treated as if it had been set to "optimum". A property may define additional constraints on the values, and additional permitted values and their semantics; e.g. 'auto' or `<percentage>`.

`<length>`

- A signed length value where a 'length' is a real number plus a unit qualification. A property may define additional constraints on the value.

`<name>`

- A string of characters representing a name. It must conform to the definition of an NCName in XML Names or XML Names 1.1.

`<number>`

- A signed real number which consists of an optional '+' or '-' character followed by a sequence of digits followed by an optional '.' character and sequence of digits. A property may define additional constraints on the value.

`<percentage>`

- A signed real percentage which consists of an optional '+' or '-' character followed by a sequence of digits followed by an optional '.' character and sequence of digits followed by '%'. A property may define additional constraints on the value.

`<script>`

- A string of characters conforming to an ISO 15924 script code.

`<shape>`

- "rect (" top= right= bottom= left= ")" where top=, bottom= right=, and left= specify offsets from the respective sides of the content rectangle of the area.
- top=, right=, bottom=, and left= may either have a `<length>` value or 'auto'. Negative lengths are permitted. The value 'auto' means that a given edge of the clipping region will be the same as the edge of the content rectangle of the area (i.e., 'auto' means the same as 'Opt'.)

`<space>`

- A compound datatype, with components: minimum, optimum, maximum, precedence, and conditionality. The minimum, optimum, and maximum components are <length>s. The precedence component is either "force" or an <integer>. The conditionality component is either "discard" or "retain". If "minimum" is greater than optimum, it will be treated as if it had been set to "optimum". If "maximum" is less than optimum, it will be treated as if it had been set to "optimum".

<string>

- A sequence of characters.

<time>

- A <number> immediately followed by a time unit identifier. Time unit identifiers are: 'ms' (for milliseconds) and 's' (for seconds).

<uri-specification>

- A sequence of characters that is "url(", followed by optional white space, followed by an optional single quote (') or double quote (") character, followed by an IRI reference as defined in RFC3987, followed by an optional single quote (') or double quote (") character, followed by optional white space, followed by ")". The two quote characters must be the same and must both be present or absent. If the IRI reference contains a single quote, the two quote characters must be present and be double quotes.

# Recommendation bibliography

Annex D - XSL-FO property summaries  
Section 2 - Data types



Many of the data types refer to other standards and recommendations:

## CSS2

- World Wide Web Consortium. *Cascading Style Sheets, level 2 (CSS2)*, as amended by Errata document 2001/04/04. W3C Recommendation.  
(<http://www.w3.org/TR/1998/REC-CSS2-19980512/>)

## DSSSL

- International Organization for Standardization, International Electrotechnical Commission. *ISO/IEC 10179:1996. Document Style Semantics and Specification Language (DSSSL)*. International Standard.

## ICC

- International Color Consortium. *Specification ICC.1:1998-09, File Format for Color Profiles*. ([http://www.color.org/ICC-1\\_1998-09.PDF](http://www.color.org/ICC-1_1998-09.PDF))

## IEEE 754

- Institute of Electrical and Electronics Engineers. *IEEE Standard for Binary Floating-Point Arithmetic*. ANSI/IEEE Std 754-1985.

## ISO15924

- International Organization for Standardization. *ISO 15924:1998. Code for the representation of names of scripts*. Draft International Standard.

## ISO31

- International Organization for Standardization. *ISO 31:1992, Amended 1998. Quantities and units* International Standard.

## ISO3166-1

- International Organization for Standardization. *ISO 3166-1:1997. Codes for the representation of names of countries and their subdivisions - Part 1: Country codes*. International Standard.

## ISO3166-2

- International Organization for Standardization. *ISO 3166-2:1998. Codes for the representation of names of countries and their subdivisions - Part 2: Country subdivision code*. International Standard.

## ISO3166-3

- International Organization for Standardization. *ISO 3166-3:1999. Codes for the representation of names of countries and their subdivisions - Part 3: Code for formerly used names of countries*. International Standard.

## ISO639

- International Organization for Standardization. *ISO 639:1998. Code for the representation of names of languages* International Standard.

## ISO639-2

- International Organization for Standardization. *ISO 639-2:1998. Codes for the representation of names of languages - Part 2: Alpha-3 code*. International Standard.

## JLS

- J. Gosling, B. Joy, and G. Steele. *The Java Language Specification*. (<http://java.sun.com/docs/books/jls/index.html>)

## OpenType

- Microsoft, Adobe. *OpenType specification v.1.2*. (<http://www.microsoft.com/truetype/tt/tt.htm>)

## RDF

- World Wide Web Consortium. *Resource Description Framework (RDF) Model and Syntax Specification*. W3C Recommendation. (<http://www.w3.org/TR/REC-rdf-syntax/>)

## RFC2070

- IETF. *RFC 2070. Internationalization of the Hypertext Markup Language*. (<http://www.ietf.org/rfc/rfc2070.txt>)

## RFC2119

- IETF. *RFC 2119. Key words for use in RFCs to Indicate Requirement Levels*. (<http://www.ietf.org/rfc/rfc2119.txt>)

## RFC3066

- IETF. *RFC 3066. Tags for the Identification of Languages*. (<http://www.ietf.org/rfc/rfc3066.txt>)

## RFC3987

- IETF. *RFC 3987. Internationalized Resource Identifiers (IRIs)*. (<http://www.ietf.org/rfc/rfc3987.txt>)

## sRGB

- Anderson, M., Motta, R., Chandrasekar, S., and Stokes, M. *A Standard Default Color Space for the Internet - sRGB*. (<http://www.w3.org/Graphics/Color/sRGB.html>)

## UNICODE Character Database

- Unicode Consortium. Unicode Character Database. (<http://www.unicode.org/Public/UNIDATA/>)

## UNICODE TR20

- Unicode Consortium. Dürst, Martin and Freytag, Asmus. *Unicode Technical Report #20. Unicode in XML and other Markup Languages* Unicode Technical Report. (<http://www.unicode.org/unicode/reports/tr20/>)

## UNICODE UAX #9

- Unicode Consortium. *The Unicode Standard, Version 3.1.0. Unicode Standard Annex #9: The Bidirectional Algorithm*. (<http://www.unicode.org/unicode/reports/tr9/>)

## XML

- World Wide Web Consortium. *Extensible Markup Language (XML) 1.0*. W3C Recommendation. (<http://www.w3.org/TR/REC-xml/>)

## XML 1.1

- World Wide Web Consortium. *Extensible Markup Language (XML) 1.1*. W3C Recommendation. (<http://www.w3.org/TR/xml11/>)

## XML Names

- World Wide Web Consortium. *Namespaces in XML*. W3C Recommendation. (<http://www.w3.org/TR/REC-xml-names/>)

## XML Names 1.1

- World Wide Web Consortium. *Namespaces in XML 1.1*. W3C Recommendation. (<http://www.w3.org/TR/xml-names11/>)

## XPath

- World Wide Web Consortium. *XML Path Language*. W3C Recommendation. (<http://www.w3.org/TR/xpath>)

## XSL 1.0

- World Wide Web Consortium. *Extensible Stylesheet Language (XSL)*. W3C Recommendation. (<http://www.w3.org/TR/2001/REC-xsl-20011015/>)

## XSL 1.1 Requirements

- World Wide Web Consortium. *XSL Requirements, Version 1.1*. W3C Recommendation. (<http://www.w3.org/TR/2003/WD-xsl11-req-20031217/>)

## XSLT

- World Wide Web Consortium. *XSL Transformations (XSLT)*. W3C Recommendation. (<http://www.w3.org/TR/xslt>)





## Format tokens from XSLT 1.0 subset

Annex D - XSL-FO property summaries  
Section 2 - Data types

A token can be used to represent how page numbers are to be rendered as a sequence of characters:

- includes borrowed definitions from XSLT 1.0

language="indication-AVT"

- indication of the language of words and letters (using same values as `xml:lang`)
- note this is named differently than for XSLT

country="indication-AVT"

- indication of the country cultural conventions

format="token-AVT"

- specifies the counting scheme to be used when formatting the value
- format="1" counts 1, 2, ..., 9, 10, 11, ..., 99, 100, 101, ...
- format="01" counts 01, 02, ..., 09, 10, ..., 99, 100, ...
  - each "0" prefix is a zero-fill indication for number values formatted less than the length of the format string
  - grouping-separator="," specifies the character between groups of digits
  - grouping-size="3" specifies the number of digits in each group (e.g.: 1,000,000)
- format="a" counts a, b, ..., z, aa, ab, ac, ...
- format="A" counts A, B, ... Z, AA, AB, AC, ...
- format="i" counts i, ii, iii, iv, v, ..., ix, x, xi, ...
- format="I" counts I, II, III, IV, V, ..., IX, X, XI, ...
- format="a-Unicode-character" specifies a translation
  - converts the number into a representation based upon a specific language, pointing to the "zero" character with a Unicode digit class
  - letter-value="alphabetic" and letter-value="traditional" for ambiguous distinctions
    - distinguishes numbering schemes in those languages where the first character of the sequence is ambiguous
    - unlike English where the differing first characters of "a" and "i" distinguish alphabetic and roman numeral formats

# Inherited properties

Annex D - XSL-FO property summaries  
Section 3 - Property summaries



The following traits are inherited from ancestral property specifications:

1 allowed-height-scale=(7.15.1;182)  
 1 allowed-width-scale=(7.15.2;182)  
 auto-restore=(7.23.2;183)  
 azimuth=(7.7.1;183)  
 border-collapse=(7.28.3;190)  
 border-separation=(7.28.5;193)  
 border-spacing=(7.31.9;193)  
 caption-side=(7.28.7;197)  
 1 change-bar-color=(7.30.2;197)  
 1 change-bar-offset=(7.30.3;197)  
 1 change-bar-placement=(7.30.4;198)  
 1 change-bar-style=(7.30.5;198)  
 1 change-bar-width=(7.30.6;198)  
 color=(7.18.1;199)  
 country=(7.10.1;200)  
 direction=(7.29.1;201)  
 display-align=(7.14.4;201)  
 elevation=(7.7.4;202)  
 empty-cells=(7.28.10;202)  
 end-indent=(7.11.8;203)  
 font=(7.31.13;204)  
 font-family=(7.9.2;204)  
 font-selection-strategy=(7.9.3;205)  
 font-size=(7.9.4;205)  
 font-size-adjust=(7.9.6;205)  
 font-stretch=(7.9.5;206)  
 font-style=(7.9.7;206)  
 font-variant=(7.9.8;206)  
 font-weight=(7.9.9;206)  
 glyph-orientation-horizontal=(7.29.2;207)  
 glyph-orientation-vertical=(7.29.3;207)  
 hyphenate=(7.10.4;208)  
 hyphenation-character=(7.10.5;208)  
 hyphenation-keep=(7.16.1;208)  
 hyphenation-ladder-count=(7.16.2;208)  
 hyphenation-push-character-count=(7.10.6;208)  
 hyphenation-remain-character-count=(7.10.7;209)  
 1 intrinsic-scale-value=(7.30.9;210)  
 intrusion-displace=(7.19.3;211)  
 keep-together=(7.20.3;211)  
 language=(7.10.2;212)

last-line-end-indent=(7.16.3;212)  
 leader-alignment=(7.22.1;212)  
 leader-length=(7.22.4;212)  
 leader-pattern=(7.22.2;212)  
 leader-pattern-width=(7.22.3;213)  
 letter-spacing=(7.17.2;213)  
 line-height=(7.16.4;213)  
 line-height-shift-adjustment=(7.16.5;214)  
 line-stacking-strategy=(7.16.6;214)  
 linefeed-treatment=(7.16.7;214)  
 ¶ merge-pages-across-index-key-references=(7.24.6;216)  
 ¶ merge-ranges-across-index-key-references=(7.24.4;217)  
 ¶ merge-sequential-page-numbers=(7.24.5;217)  
 orphans=(7.20.6;218)  
 page-break-inside=(7.31.18;221)  
 ¶ page-number-treatment=(7.24.3;222)  
 pitch=(7.7.7;223)  
 pitch-range=(7.7.8;223)  
 provisional-distance-between-starts=(7.30.12;224)  
 provisional-label-separation=(7.30.11;225)  
 relative-align=(7.14.6;226)  
 richness=(7.7.10;227)  
 rule-style=(7.22.5;228)  
 rule-thickness=(7.22.6;228)  
 ¶ scale-option=(7.30.14;228)  
 score-spaces=(7.30.15;229)  
 script=(7.10.3;229)  
 speak=(7.7.11;231)  
 speak-header=(7.7.12;231)  
 speak-numeral=(7.7.13;231)  
 speak-punctuation=(7.7.14;231)  
 speech-rate=(7.7.15;232)  
 start-indent=(7.11.7;232)  
 stress=(7.7.16;233)  
 text-align=(7.16.9;234)  
 text-align-last=(7.16.10;234)  
 text-indent=(7.16.11;235)  
 text-transform=(7.17.6;235)  
 visibility=(7.30.17;236)  
 voice-family=(7.7.17;237)  
 volume=(7.7.18;237)  
 white-space=(7.31.23;237)  
 white-space-collapse=(7.16.12;237)  
 white-space-treatment=(7.16.8;238)  
 widows=(7.20.7;238)  
 word-spacing=(7.17.8;238)

wrap-option=(7.16.13;238)  
writing-mode=(7.29.7;238)  
❏ xml:lang=(7.31.24;239)

PREVIEW EXCERPT



# Shorthand properties

Annex D - XSL-FO property summaries  
Section 3 - Property summaries

## Important notes:

- shorthand properties need not be supported by a processor unless it is claiming complete conformance (the highest XSL-FO conformance level)
  - this is an important portability issue in that if one develops a stylesheet using a formatter that does recognize shorthand properties, that stylesheet may not run properly on a formatter that does not recognize shorthand properties
- if the value "inherit" is being used for a shorthand, it applies to all subproperty values
  - other subproperty values cannot be specified at the same time

## Summary of shorthand properties

background=(7.31.1;183)  
 background-position=(7.31.2;184)  
 border=(7.31.3;186)  
 border-bottom=(7.31.4;189)  
 border-color=(7.31.5;190)  
 border-left=(7.31.6;191)  
 border-right=(7.31.7;192)  
 border-spacing=(7.31.9;193)  
 border-style=(7.31.8;195)  
 border-top=(7.31.10;195)  
 border-width=(7.31.11;196)  
 cue=(7.31.12;200)  
 font=(7.31.13;204)  
 margin=(7.31.14;214)  
 padding=(7.31.15;218)  
 page-break-after=(7.31.16;221)  
 page-break-before=(7.31.17;221)  
 page-break-inside=(7.31.18;221)  
 pause=(7.31.19;222)  
 position=(7.31.20;224)  
 size=(7.31.21;229)  
 vertical-align=(7.31.22;236)  
 white-space=(7.31.23;237)  
 ¶ xml:lang=(7.31.24;239)

## Property summary

Annex D - XSL-FO property summaries  
Section 3 - Property summaries



---

A alphabetical listing of all properties indicating each of:

- section number where found in Recommendation
- whether originally from CSS or from XSL
- conformance level (basic, extended, complete) required for support to be expected

PREVIEW EXCERPT

## Property definition notes:

- the following summary is extracted from the XSL Recommendation and that references to "prose" refer to the prose found in the Recommendation as noted in the given section number, and not necessarily to the content of this book
- in certain cases the XSL Recommendation lists the applicable formatting objects indirectly through the CSS definition and the list is incorrect due to XSL overrides
  - in this reference that list has been replaced with a synthesized list based on explicit references made in the XSL Recommendation definition of the formatting to the property
    - directly for the object
    - indirectly through common properties
    - indirectly through shorthand properties whose

`absolute-position`= (7.6.1; CSS; complete):

- Value: `auto` | `absolute` | `fixed` | `inherit`
- Initial: `auto`
- specifies an object's position as normally flowed relative to its sibling ("`auto`"), offset from its containing area ("`absolute`"), offset from the medium ("`fixed`"), or that of an ascendent specified value ("`inherit`")
- object to which this property applies: `<block-container>`
- shorthand impacting on this property: `position`=
- Media: visual

`active-state`= (7.23.1; XSL; extended):

- Value: `link` | `visited` | `active` | `hover` | `focus`
- Initial: `no`, a value is required
- the state tested for a `<basic-link>` descendant of the parent `<multi-properties>` as being either not yet visited ("`link`"), ready to be engaged ("`hover`"), in the act of being engaged ("`active`"), the focus for input ("`focus`"), and already having been visited ("`visited`")
- object to which this property applies: `<multi-property-set>`
- Media: interactive

`alignment-adjust`= (7.14.1; XSL; basic):

- Value: `auto` | `baseline` | `before-edge` | `text-before-edge` | `middle` | `central` | `after-edge` | `text-after-edge` | `ideographic` | `alphabetic` | `hanging` | `mathematical` | `<percentage>` | `<length>` | `inherit`
- Initial: `auto`
- Percentages: see prose
- Media: visual

- specifies the alignment point on the object with which the formatter aligns areas to the alignment point on the line as specified by the `alignment-baseline=` setting
  - refer to `baseline-shift=` for suitable (not exclusive) behavior for text
- percentages are of either the computed area for an image, the font size for a character, or the line height for other constructs
- positive lengths are opposite to the shift direction
- see Chapter 4 Area and page basics (page 76) for details of values
- objects to which this property applies: `<basic-link>`, `<character>`, `<external-graphic>`, `<inline>`, `<inline-container>`, `<instream-foreign-object>`, `<leader>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`
- shorthand impacting on this property: `vertical-align=`

`alignment-baseline=` (7.14.2; XSL; basic):

- Value: `auto` | `baseline` | `before-edge` | `text-before-edge` | `middle` | `central` | `after-edge` | `text-after-edge` | `ideographic` | `alphabetic` | `hanging` | `mathematical` | `inherit`
- Initial: `auto`
- specifies the alignment point on the line with which the formatter aligns areas to the alignment point on the object as specified by the `alignment-adjust=` setting
- objects to which this property applies: `<basic-link>`, `<character>`, `<external-graphic>`, `<inline>`, `<inline-container>`, `<instream-foreign-object>`, `<leader>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`
- shorthand impacting on this property: `vertical-align=`

- Media: visual

① `allowed-height-scale=` (7.15.1; XSL; extended):

- Value: [ `any` | `<percentage>` ]\* | `inherit`
- Initial: `any`
- an unordered white-space-separated list of percentages and possibly the token "any" to constrain the possibly choices of height scaling factors to be applied to images
- there should be an "any" value in the list to prevent the formatter from choosing a scaling factor that might be larger than the viewport
- objects to which this property applies: `<external-graphic>`, `<instream-foreign-object>`

- Percentages: intrinsic height

- Media: visual

① `allowed-width-scale=` (7.15.2; XSL; extended):

- Value: [ `any` | `<percentage>` ]\* | `inherit`
- Initial: `any`

- Percentages: intrinsic width

- Media: visual



- an unordered white-space-separated list of percentages and possibly the token "any" to constrain the possible choices of width scaling factors to be applied to images
- there should be an "any" value in the list to prevent the formatter from choosing a scaling factor that might be larger than the viewport
- objects to which this property applies: <external-graphic>, <instream-foreign-object>

auto-restore= (7.23.2; XSL; extended):

- Value: true | false
- Initial: false
- when "true" for an object that object's <multi-case> will be restored to its initial value when that object is hidden by the hiding of an ancestral <multi-switch> construct
- object to which this property applies: <multi-switch>

azimuth= (7.7.1; CSS; basic):

- Value: <angle> | [[ left-side | far-left | left | center-left | center | center-right | right | far-right | right-side ] || behind ] | leftwards | rightwards | inherit
- Initial: center
- directs from which lateral direction an aural presentation is heard with respect to the listener
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

background= (7.31.1; CSS; shorthand):

- Value: [background-color= || background-image= || background-repeat= || background-attachment= || background-position= ] | inherit
- Initial: not defined for shorthand properties
- provides a shorthand method of specifying the five individual values
- any values omitted are given their initial values

background-attachment= (7.8.1; CSS; extended):

- Value: scroll | fixed | inherit
- Initial: scroll

- when the background is "fixed" it remains stationary in the viewport while the foreground scrolls in front of it
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: background=

background-color= (7.8.2; CSS; basic):

- Value: <color> | transparent | inherit
- Initial: transparent
- Media: visual
- fills the padding rectangle behind an object's content, thus the color is seen within the inside edge of a visible border
  - gaps in a visible border show through the background color of the parent area
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: background=

background-image= (7.8.3; CSS; extended):

- Value: <uri-specification> | none | inherit
- Initial: none
- Media: visual
- positioned (and possibly repeated with background-repeat= within the padding rectangle behind an object's content
- the background-color= will show through transparent parts of the background image
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: background=

background-position= (7.31.2; CSS; shorthand):

- Value: [ [<percentage> | <length> ]{1,2} ] [ [top | center | bottom] || [left | center | right] ] | inherit
- Initial: 0% 0%
- Percentages: refer to the size of the box itself
- Media: visual

- a shorthand for specifying one or two positions oriented to the upper left corner of the padding-rectangle, setting individual values for background-position-horizontal= and background-position-vertical=
- the horizontal value is specified first followed by the vertical value
- an absent value is interpreted as "center" which is placed at a position of 50%

background-position-horizontal= (7.8.5; CSS; extended):

- Value: <percentage> | <length> | left | center | right | inherit
- Initial: 0%
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: background-position=

background-position-vertical= (7.8.6; CSS; extended):

- Value: <percentage> | <length> | top | center | bottom | inherit
- Initial: 0%
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: background-position=

background-repeat= (7.8.4; CSS; extended):

- Value: repeat | repeat-x | repeat-y | no-repeat | inherit
- Initial: repeat
- Media: visual

- "x" is the horizontal direction, "y" is the vertical direction, both relative to the reference-orientation but independent of the writing-mode
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: background=

baseline-shift= (7.14.3; XSL; basic):

- Value: baseline | sub | super | <percentage> | <length> | inherit
- Initial: baseline
- repositions the dominant-baseline (and all other entries in the baseline-table) of an object relative to the dominant-baseline of the parent area
- using the parent area as a reference is more consistent for a set of child objects of different sizes
- objects to which this property applies: <basic-link>, <character>, <external-graphic>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>
- shorthand impacting on this property: vertical-align=
- Percentages: refers to the "line-height" of the parent area
- Media: visual

blank-or-not-blank= (7.27.1; XSL; extended):

- Value: blank | not-blank | any | inherit
- Initial: any
- object to which this property applies: <conditional-page-master-reference>
- Media: visual

block-progression-dimension= (7.15.3; CSS; basic):

- Value: auto | <length> | <percentage> | <length-range> | inherit
- Initial: auto
- specifies the block-progression-dimension of the content-rectangle for each area generated by the formatting object
- objects to which this property applies: <block-container>, <external-graphic>, <inline>, <inline-container>, <instream-foreign-object>, <table>, <table-caption>, <table-cell>, <table-row>
- Percentages: see prose
- Media: visual

border= (7.31.3; CSS; shorthand):

- Value: [ border-width= || border-style= || [ <color> | transparent ] ] | inherit
- Initial: see individual properties
- sets any of the border width, style or color properties for all of the four borders of a box
- Media: visual

border-after-color= (7.8.10; CSS; basic):

- Value: <color> | transparent | inherit - Media: visual
- Initial: the value of the 'color' property
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border=, border-color=

border-after-precedence= (7.28.1; XSL; basic):

- Value: force | <integer> | inherit - Media: visual
- Initial: <table>: 6, <table-cell>: 5, <table-column>: 4, <table-row>: 3, <table-body>: 2, <table-header>: 1, <table-footer>: 0
- specifies the precedence of the formatting object's border compared to the precedence of the coincident border of:
  - an adjacent formatting object when borders are collapsed using border-collapse=
  - an overlapping table construct in the formatting object hierarchy
- the border specification precedence that is highest of all of the coincident borders dictates which formatting object's properties that border will exhibit
- objects to which this property applies: <table>, <table-body>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>

border-after-style= (7.8.11; CSS; basic):

- Value: border-style= | inherit - Media: visual
- Initial: none
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: border-style=

border-after-width= (7.8.12; CSS; basic):

- Value: border-width= | <length-conditional> | inherit - Media: visual
- Initial: medium

- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: border-width=

border-before-color= (7.8.7; CSS; basic):

- Value: <color> | transparent | inherit
- Initial: the value of the 'color' property
- Media: visual
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border=, border-color=

border-before-precedence= (7.28.2; XSL; basic):

- Value: force | <integer> | inherit
- Initial: <table>: 6, <table-cell>: 5, <table-column>: 4, <table-row>: 3, <table-body>: 2, <table-header>: 1, <table-footer>: 0
- Media: visual
- see border-after-precedence= for details
- objects to which this property applies: <table>, <table-body>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>

border-before-style= (7.8.8; CSS; basic):

- Value: border-style= | inherit
- Initial: none
- Media: visual
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: border-style=

border-before-width= (7.8.9; CSS; basic):

- Media: visual

- Value: border-width= | <length-conditional> | inherit
- Initial: medium
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: border-width=

border-bottom= (7.31.4; CSS; shorthand):

- Value: [ border-width= || border-style= || [ <color> | transparent ] ] | inherit
- Media: visual
- Initial: see individual properties
- a shorthand for specifying a border's width, style and color

border-bottom-color= (7.8.22; CSS; basic):

- Value: <color> | transparent | inherit
- Media: visual
- Initial: the value of the 'color' property
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-bottom=, border-color=

border-bottom-style= (7.8.23; CSS; basic):

- Value: border-style= | inherit
- Media: visual
- Initial: none
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-bottom=, border-style=

border-bottom-width= (7.8.24; CSS; basic):

- Value: border-width= | inherit
- Media: visual
- Initial: medium

- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-bottom=, border-width=

border-collapse= (7.28.3; CSS; extended):

- Value: collapse | collapse-with-precedence | separate | inherit
- Initial: collapse
- object to which this property applies: <table>
- Media: visual

border-color= (7.31.5; CSS; shorthand):

- Value: [ <color> | transparent ]{1,4} | inherit
- Initial: see individual properties
- up to four individual values for the border color can be specified in this shorthand
  - see Chapter 10 Breaks, keeps, spacing, borders and backgrounds (page 114) for a discussion of borders
- four colors specifies each of the borders of a table in the order: before, end, after, start
- three colors specifies the borders in the order: before, end/start, after
- two colors specifies the borders in the order: before/after, end/start
- one color specifies the color of all borders
- Media: visual

border-end-color= (7.8.16; CSS; basic):

- Value: <color> | transparent | inherit
- Initial: the value of the 'color' property
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border=, border-color=
- Media: visual

border-end-precedence= (7.28.4; XSL; basic):

- Value: force | <integer> | inherit
- Initial: <table>: 6, <table-cell>: 5, <table-column>: 4, <table-row>: 3, <table-body>: 2, <table-header>: 1, <table-footer>: 0
- Media: visual



- see border-after-precedence= for details
- objects to which this property applies: <table>, <table-body>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>

border-end-style= (7.8.17; CSS; basic):

- Value: border-style= | inherit
- Initial: none
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: border-style=

- Media: visual

border-end-width= (7.8.18; CSS; basic):

- Value: border-width= | <length-conditional> | inherit
- Initial: medium
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: border-width=

- Media: visual

border-left= (7.31.6; CSS; shorthand):

- Value: [ border-width= || border-style= || [ <color> | transparent ] ] | inherit
- Initial: see individual properties
- a shorthand for specifying a border's width, style and color

- Media: visual

border-left-color= (7.8.25; CSS; basic):

- Value: <color> | transparent | inherit
- Initial: the value of the 'color' property

- Media: visual

- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-color=, border-left=

border-left-style= (7.8.26; CSS; basic):

- Value: border-style= | inherit
- Media: visual
- Initial: none
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-left=, border-style=

border-left-width= (7.8.27; CSS; basic):

- Value: border-width= | inherit
- Media: visual
- Initial: medium
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-left=, border-width=

border-right= (7.31.7; CSS; shorthand):

- Value: [ border-width= || border-style= || [ <color> | transparent ] ] | inherit
- Media: visual
- Initial: see individual properties
- a shorthand for specifying a border's width, style and color

border-right-color= (7.8.28; CSS; basic):

- Value: <color> | transparent | inherit
- Media: visual
- Initial: the value of the 'color' property

- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-color=, border-right=

border-right-style= (7.8.29; CSS; basic):

- Value: border-style= | inherit
- Media: visual
- Initial: none
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-right=, border-style=

border-right-width= (7.8.30; CSS; basic):

- Value: border-width= | inherit
- Media: visual
- Initial: medium
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-right=, border-width=

border-separation= (7.28.5; XSL; extended):

- Value: <length-bp-ip-direction> | inherit
- Media: visual
- Initial: .block-progression-direction="0pt" .inline-progression-direction="0pt"
- specifies the distance between the borders of adjacent cells when using borders are not collapsed using border-collapse=
- the space in between borders is filled with the table background color
- object to which this property applies: <table>
- shorthand impacting on this property: border-spacing=

border-spacing= (7.31.9; CSS; shorthand):

- Media: visual

- Value: <length> <length>? | inherit
- Initial: Opt
- this shorthand specifies the two components of the compound border-separation= property as separate values in the order of the .inline-progression-direction component from the first value then the .block-progression-direction component from the second value
- specifying only one value sets both components simultaneously

border-start-color= (7.8.13; CSS; basic):

- Value: <color> | transparent | inherit
- Initial: the value of the 'color' property
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border=, border-color=
- Media: visual

border-start-precedence= (7.28.6; XSL; basic):

- Value: force | <integer> | inherit
- Initial: <table>: 6, <table-cell>: 5, <table-column>: 4, <table-row>: 3, <table-body>: 2, <table-header>: 1, <table-footer>: 0
- see border-after-precedence= for details
- objects to which this property applies: <table>, <table-body>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>
- Media: visual

border-start-style= (7.8.14; CSS; basic):

- Value: border-style= | inherit
- Initial: none
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: border-style=
- Media: visual

border-start-width= (7.8.15; CSS; basic):

- Value: border-width= | <length-conditional> | inherit
- Initial: medium
- Media: visual

- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: border-width=

border-style= (7.31.8; CSS; shorthand):

- Value: border-style{ 1,4 } | inherit
- Media: visual
- Initial: see individual properties
- up to four individual values for the border style can be specified in this shorthand
  - see Chapter 10 Breaks, keeps, spacing, borders and backgrounds (page 114) for a discussion of borders
- four values specifies each of the borders of a table in the order: before, end, after, start
- three values specifies the borders in the order: before, end/start, after
- two values specifies the borders in the order: before/after, end/start
- one value specifies the style of all borders

border-top= (7.31.10; CSS; shorthand):

- Value: [ border-width= || border-style= || [ <color> | transparent ] ] | inherit
- Media: visual
- Initial: see individual properties
- a shorthand for specifying a border's width, style and color

border-top-color= (7.8.19; CSS; basic):

- Value: <color> | transparent | inherit
- Media: visual
- Initial: the value of the 'color' property
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-color=, border-top=

border-top-style= (7.8.20; CSS; basic):

- Value: border-style= | inherit
- Media: visual
- Initial: none

- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-style=, border-top=

border-top-width= (7.8.21; CSS; basic):

- Value: border-width= | inherit
- Initial: medium
- Media: visual
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthands impacting on this property: border-top=, border-width=

border-width= (7.31.11; CSS; shorthand):

- Value: border-width{ 1,4 } | inherit
- Initial: see individual properties
- Media: visual
- up to four individual values for the border width can be specified in this shorthand
  - see Chapter 10 Breaks, keeps, spacing, borders and backgrounds (page 114) for a discussion of borders
- four values specifies each of the borders of a table in the order: before, end, after, start
- three values specifies the borders in the order: before, end/start, after
- two values specifies the borders in the order: before/after, end/start
- one value specifies the width of all borders

bottom= (7.6.4; CSS; extended):

- Value: <length> | <percentage> | auto | inherit
- Initial: auto
- Percentages: refer to height of containing block
- Media: visual
- specifies the distance between the margin edge and the containing block
- a value other than "auto" overrides a height= value of "auto"
- the percentage is ignored if the containing block height is not specified explicitly
- object to which this property applies: <block-container>

break-after= (7.20.1; XSL; basic):

- Value: auto | column | page | even-page | odd-page | inherit
- Initial: auto
- Media: visual

- this property has no effect on `<table-row>` if that row has a row-spanning cell including the following row
- objects to which this property applies: `<block>`, `<block-container>`, `<list-block>`, `<list-item>`, `<table>`, `<table-and-caption>`, `<table-row>`
- shorthand impacting on this property: `page-break-after=`

`break-before=` (7.20.2; XSL; basic):

- Value: `auto` | `column` | `page` | `even-page` | `odd-page` - Media: visual  
| `inherit`
- Initial: `auto`
- this property has no effect on `<table-row>` if the previous row has a row-spanning cell including the given row
- objects to which this property applies: `<block>`, `<block-container>`, `<list-block>`, `<list-item>`, `<table>`, `<table-and-caption>`, `<table-row>`
- shorthand impacting on this property: `page-break-before=`

`caption-side=` (7.28.7; CSS; complete):

- Value: `before` | `after` | `start` | `end` | `top` | `bottom` | `left` - Media: visual  
| `right` | `inherit`
- Initial: `before`
- object to which this property applies: `<table-and-caption>`

`case-name=` (7.23.3; XSL; extended):

- Value: `<name>` - Media: interactive
- Initial: none, a value is required
- this name must be unique among the siblings of the `<multi-case>`
- object to which this property applies: `<multi-case>`

`case-title=` (7.23.4; XSL; extended):

- Value: `<string>` - Media: interactive
- Initial: none, a value is required
- this string can be displayed in a menu corresponding to the associated `<multi-case>` objects of allowed `<multi-toggle>` destinations when more than one candidate destination is specified
- object to which this property applies: `<multi-case>`

1 `change-bar-class=` (7.30.1; XSL; extended):

- Value: `<name>` - Media: visual
- Initial: none, value required
- a name used when pairing the beginning and end of a change bar
- objects to which this property applies: `<change-bar-begin>`, `<change-bar-end>`

1 `change-bar-color=` (7.30.2; XSL; extended):

- Value: `<color>` - Media: visual
- Initial: the value of the color property
- the color used throughout a change bar
- object to which this property applies: `<change-bar-begin>`

1 `change-bar-offset=` (7.30.3; XSL; extended):

- Value: <length>
- Initial: 6pt
- the distance from the edge of the column area and the center of the change bar
- object to which this property applies: <change-bar-begin>

④ change-bar-placement= (7.30.4; XSL; extended):

- Value: start | end | left | right | inside | outside | alternate
- Initial: start
- positions the change bar relative to the column areas
- object to which this property applies: <change-bar-begin>

④ change-bar-style= (7.30.5; XSL; extended):

- Value: border-style=
- Initial: none
- the style to use for the change bar, using the definitions as for borders (see Chapter 10 Breaks, keeps, spacing, borders and backgrounds (page 114))
- object to which this property applies: <change-bar-begin>

④ change-bar-width= (7.30.6; XSL; extended):

- Value: border-width=
- Initial: medium
- the width to use for the change bar, using the definitions as for borders (see Chapter 10 Breaks, keeps, spacing, borders and backgrounds (page 114))
- object to which this property applies: <change-bar-begin>

character= (7.17.1; XSL; basic):

- Value: <character>
- Initial: N/A, value is required
- object to which this property applies: <character>

clear= (7.19.1; CSS; extended):

- Value: start | end | left | right | inside | outside | both | none | inherit
- Initial: none
- specifies which side-floats whose parent reference-area is the nearest ancestor of the reference-area of the generated area must be clear of the given block to which this property is specified
  - the value indicates which side float the given construct is supposed to clear
- this may alter the space-before= property to meet the constraints
- also applies to <block> even though not listed in the Recommendation in the list of properties for the object
- objects to which this property applies: <block>, <block-container>, <float>, <list-block>, <table>, <table-and-caption>

clip= (7.21.1; CSS; extended):

- Value: <shape> | auto | inherit
- Initial: auto



- the only shape supported is `rect( top, right, bottom, left )`
  - the four values are the offsets of the clipping area from the respective sides of the containing area
- the value "auto" represents an offset of zero
- objects to which this property applies: `<block-container>`, `<external-graphic>`, `<inline-container>`, `<instream-foreign-object>`, `<region-after>`, `<region-before>`, `<region-body>`, `<region-end>`, `<region-start>`

`color=` (7.18.1; CSS; basic):

- Value: `<color>` | inherit
- Initial: depends on user agent
- see Function groupings (page 136) for the color functions available
- objects to which this property applies: `<bidirectional-override>`, `<block>`, `<bookmark-title>`, `<character>`, `<initial-property-set>`, `<inline>`, `<leader>`, `<title>`

- Media: visual

`color-profile-name=` (7.18.2; XSL; extended):

- Value: `<name>` | inherit
- Initial: N/A, value is required
- object to which this property applies: `<color-profile>`

- Media: visual

`column-count=` (7.27.2; XSL; extended):

- Value: `<number>` | inherit
- Initial: 1
- object to which this property applies: `<region-body>`

- Media: visual

`column-gap=` (7.27.3; XSL; extended):

- Value: `<length>` | `<percentage>` | inherit
- Initial: 12.0pt

- Percentages: refer to width of the region being divided into columns.

- Media: visual

- a negative value is translated to 0pt
- object to which this property applies: `<region-body>`

`column-number=` (7.28.8; XSL; basic):

- Value: `<number>`
- Initial: see prose
- the initial value for `<table-column>` is 1 plus the `column-number=` of the previous `<table-column>`, or "1" for the first
- the initial value for `<table-cell>` is the `column-number=` of the previous cell plus the `number-columns-spanned=` of that previous cell
- objects to which this property applies: `<table-cell>`, `<table-column>`

- Media: visual

`column-width=` (7.28.9; XSL; basic):

- Value: `<length>` | `<percentage>`
- Initial: see prose

- Percentages: refer to width of table

- Media: visual

- see Chapter 6 Tables (page 88) for a discussion of column widths
- object to which this property applies: `<table-column>`

content-height= (7.15.4; XSL; extended):

- Value: auto | scale-to-fit | scale-down-to-fit | scale-up-to-fit | <length> | <percentage> | inherit
- Initial: auto
- objects to which this property applies: <external-graphic>, <instream-foreign-object>
- Percentages: intrinsic height
- Media: visual

content-type= (7.30.7; XSL; extended):

- Value: <string> | auto
- Initial: auto
- the string is prefixed by "namespace-prefix:" for a namespace specification
  - a null prefix refers to the default namespace
- the string is prefixed by "content-type:" for a mime-type specification
- objects to which this property applies: <external-graphic>, <instream-foreign-object>
- Media: visual

content-width= (7.15.5; XSL; extended):

- Value: auto | scale-to-fit | scale-down-to-fit | scale-up-to-fit | <length> | <percentage> | inherit
- Initial: auto
- objects to which this property applies: <external-graphic>, <instream-foreign-object>
- Percentages: intrinsic width
- Media: visual

country= (7.10.1; XSL; extended):

- Value: none | <country> | inherit
- Initial: none
- specifies the country used in language- and locale- coupled services
  - e.g. line-justification, line-breaking, hyphenation, etc.
- objects to which this property applies: <block>, <character>, <page-sequence>, <scaling-value-citation>
- shorthand impacting on this property: xml:lang=
- Media: visual

cue= (7.31.12; CSS; shorthand):

- Value: cue-before= || cue-after= | inherit
- Initial: not defined for shorthand properties
- when a single value is specified it is applied to both properties
- Media: aural

cue-after= (7.7.2; CSS; basic):

- Value: <uri-specification> | none | inherit
- Initial: none
- Media: aural

- specifies the URL of rendered information as an "auditory icon" after a construct is rendered
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: cue=

cue-before= (7.7.3; CSS; basic):

- Value: <uri-specification> | none | inherit
- Initial: none
- Media: aural
- specifies the URL of rendered information as an "auditory icon" before a construct is rendered
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: cue=

destination-placement-offset= (7.23.5; XSL; extended):

- Value: <length>
- Initial: 0pt
- Media: interactive
- specifies where in the destination view port or page (from the beginning) that the targeted location is to be rendered
- object to which this property applies: <basic-link>

direction= (7.29.1; CSS; basic):

- Value: ltr | rtl | inherit
- Initial: ltr
- Media: visual
- this property is deprecated for all formatting objects other than <bidirectional-override>
  - if used for such other objects, this overrides the direction implied by any writing-mode= property
- object to which this property applies: <bidirectional-override>

display-align= (7.14.4; XSL; extended):

- Value: auto | before | center | after | inherit
- Initial: auto
- Media: visual

- this specifies the alignment, in the block-progression-direction, of the areas that are the children of a reference-area
- the value of "auto" infers either the `relative-align` property if applicable, or the value "before"
- objects to which this property applies: `<block-container>`, `<external-graphic>`, `<inline-container>`, `<instream-foreign-object>`, `<region-after>`, `<region-before>`, `<region-body>`, `<region-end>`, `<region-start>`, `<table-cell>`

`dominant-baseline`= (7.14.5; XSL; basic):

- Value: `auto` | `use-script` | `no-change` | `reset-size` | `ideographic` | `alphabetic` | `hanging` | `mathematical` | `central` | `middle` | `text-after-edge` | `text-before-edge` | `inherit`
- Initial: `auto`
- determines or re-determines a `scaled-baseline-table`
- objects to which this property applies: `<basic-link>`, `<character>`, `<external-graphic>`, `<inline>`, `<inline-container>`, `<instream-foreign-object>`, `<leader>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`
- shorthand impacting on this property: `vertical-align`
- Media: visual

`elevation`= (7.7.4; CSS; basic):

- Value: `<angle>` | `below` | `level` | `above` | `higher` | `lower` | `inherit`
- Initial: `level`
- the angles "90deg" and "-90deg" position the sound respectively directly above or below the listener
- the values "higher" and "lower" respectively increase and decrease the current elevation by 10 degrees
- objects to which this property applies: `<basic-link>`, `<bidirectional-override>`, `<block>`, `<change-bar-begin>`, `<change-bar-end>`, `<character>`, `<external-graphic>`, `<initial-property-set>`, `<inline>`, `<instream-foreign-object>`, `<leader>`, `<list-block>`, `<list-item>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`, `<table>`, `<table-and-caption>`, `<table-body>`, `<table-caption>`, `<table-cell>`, `<table-footer>`, `<table-header>`, `<table-row>`, `<title>`
- Media: aural

`empty-cells`= (7.28.10; CSS; extended):

- Value: `show` | `hide` | `inherit`
- Initial: `show`
- Media: visual

- specifies the rendering of the borders and background of cells without visible content when using the separated borders model specified by `border-collapse=`
- if all cells in a row have a value of "hide", the row behaves as if were not displayed at all
- visible content is everything other than XML white-space (carriage-return, linefeed, tab, and space)
- object to which this property applies: `<table-cell>`

`end-indent=` (7.11.8; XSL; basic):

- Value: `<length>` | `<percentage>` | `inherit`
- Initial: `0pt`
- Percentages: refer to inline-progression-dimension of containing reference-area
- Media: visual
- objects to which this property applies: `<block>`, `<block-container>`, `<list-block>`, `<list-item>`, `<region-body>`, `<simple-page-master>`, `<table>`, `<table-and-caption>`

`ends-row=` (7.28.11; XSL; extended):

- Value: `true` | `false`
- Initial: `false`
- Media: visual
- object to which this property applies: `<table-cell>`

`extent=` (7.27.4; XSL; extended):

- Value: `<length>` | `<percentage>` | `inherit`
- Initial: `0.0pt`
- Percentages: refer to the corresponding block-progression-dimension or inline-progression-dimension of the page-view-viewport-area.
- Media: visual

- objects to which this property applies: `<region-after>`, `<region-before>`, `<region-end>`, `<region-start>`

`external-destination=` (7.23.6; XSL; extended):

- Value: empty string | `<uri-specification>`
- Initial: empty string
- Media: interactive
- at least one of `external-destination=` and `internal-destination=` properties should be assigned
- if both are assigned, the system may either report the error, or use `internal-destination=`
- objects to which this property applies: `<basic-link>`, `<bookmark>`

`float=` (7.19.2; CSS; extended):

- Value: `before` | `start` | `end` | `left` | `right` | `inside` | `outside` | `none` | `inherit`
- Initial: `none`
- Media: visual
- unlike CSS, this property only applies to `<float>` and to no other formatting object
- object to which this property applies: `<float>`

`flow-map-name=` (7.27.18; XSL; extended):

- Value: <name>
- Initial: none, a value is required
- provides an identity for a flow map so that it can be referenced in a page sequence
- object to which this property applies: <flow-map>

❶ flow-map-reference= (7.27.19; XSL; extended):

- Value: <name>
- Initial: see prose
- the name of the flow map to be engaged in a page sequence
- object to which this property applies: <page-sequence>

flow-name= (7.27.5; XSL; basic):

- Value: <name>
- Initial: an empty name
- must be unique within a <page-sequence>
- the following names are reserved: "xsl-region-body", "xsl-region-before", "xsl-region-after", "xsl-region-start", "xsl-region-end", "xsl-before-float-separator", "xsl-footnote-separator"
- objects to which this property applies: <flow>, <static-content>

❶ flow-name-reference= (7.27.20; XSL; extended):

- Value: <name>
- Initial: none, a value is required
- the name of a flow to be included in a flow map
- object to which this property applies: <flow-name-specifier>

font= (7.31.13; CSS; shorthand):

- Value: [ [ font-style= || font-variant= || font-weight= ]? font-size= [ / line-height= ]? font-family= ] | caption | icon | menu | message-box | small-caption | status-bar | inherit
- Initial: see individual properties
- individual system font characteristics, such as font-family, font-size, etc. may be obtained by the use of the `system-font()` function without any arguments

font-family= (7.9.2; CSS; basic):

- Value: [[ <family-name> | *generic-family* ],]\*
- Initial: depends on user agent

- font family names with spaces should be quoted, but will otherwise be normalized
- generic font family names are not quoted
  - the values are "serif", "sans-serif", "cursive", "fantasy", and "monospace"
- this property is a prioritized list of font family names which are tried in sequence to find an available font that matches the selection criteria specified by `font-selection-strategy=`
- objects to which this property applies: `<bidirectional-override>`, `<block>`, `<character>`, `<initial-property-set>`, `<inline>`, `<leader>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`, `<title>`
- shorthand impacting on this property: `font=`

`font-selection-strategy=` (7.9.3; XSL; complete):

- Value: `auto` | `character-by-character` | `inherit` - Media: visual
- Initial: `auto`
- specifies if the font selected is done in an implementation-defined manner or if each individual character is checked for the desired font
- font selection is based on `font-family=`, `font-style=`, `font-variant=`, `font-weight=`, `font-stretch=`, `font-size=` and possibly one or more characters in context
- objects to which this property applies: `<bidirectional-override>`, `<block>`, `<character>`, `<initial-property-set>`, `<inline>`, `<leader>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`, `<title>`

`font-size=` (7.9.4; CSS; basic):

- Value: *absolute-size* | *relative-size* | `<length>` | `<percentage>` | `inherit` - Percentages: refer to parent element's font size
- Initial: `medium` - Media: visual
- possible values for *absolute-size* are: `"xx-small"`, `"x-small"`, `"small"`, `"medium"`, `"large"`, `"x-large"` and `"xx-large"`
- possible values for *relative-size* are: `"larger"` and `"smaller"`
- objects to which this property applies: `<bidirectional-override>`, `<block>`, `<character>`, `<initial-property-set>`, `<inline>`, `<leader>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`, `<title>`
- shorthand impacting on this property: `font=`

`font-size-adjust=` (7.9.6; CSS; extended):

- Value: `<number>` | `none` | `inherit` - Media: visual
- Initial: `none`

- a number value specifies the aspect (font size divided by x-height)
  - smaller fonts with larger values are typically easier to read than those with smaller values
- note that child elements will inherit unadjusted values because inheritance is based on computed values
- objects to which this property applies: <bidirectional-override>, <block>, <character>, <initial-property-set>, <inline>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <title>

font-stretch= (7.9.5; CSS; extended):

- Value: normal | wider | narrower | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded | inherit
- Initial: normal
- selects a normal, condensed or extended face from a font family
- values of "wider" and "narrower" do not go beyond the respective values of "ultra-expanded" and "ultra-condensed"
- objects to which this property applies: <bidirectional-override>, <block>, <character>, <initial-property-set>, <inline>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <title>

font-style= (7.9.7; CSS; basic):

- Value: normal | italic | oblique | backslant | inherit
- Initial: normal
- note that "italic" will match "oblique" if no italic face is available in the font family
- objects to which this property applies: <bidirectional-override>, <block>, <bookmark-title>, <character>, <initial-property-set>, <inline>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <title>
- shorthand impacting on this property: font=

font-variant= (7.9.8; CSS; basic):

- Value: normal | small-caps | inherit
- Initial: normal
- objects to which this property applies: <bidirectional-override>, <block>, <character>, <initial-property-set>, <inline>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <title>
- shorthand impacting on this property: font=

font-weight= (7.9.9; CSS; basic):

- Value: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit
- Initial: normal



- "normal" maps to a value of "400"
- "500" will map to a medium weight value (if available for the font)
- "bold" maps to a value of "700"
- "lighter" and "bolder" will find the closest font that is respectively different from the inherited font weight
- objects to which this property applies: <bidirectional-override>, <block>, <bookmark-title>, <character>, <initial-property-set>, <inline>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <title>
- shorthand impacting on this property: font=

force-page-count= (7.27.6; XSL; extended):

- Value: auto | even | odd | end-on-even | end-on-odd - Media: visual  
| no-force | inherit
- Initial: auto
- object to which this property applies: <page-sequence>

format= (7.26.1; XSL; basic):

- Value: <string> - Media: all
- Initial: 1
- see Format tokens from XSLT 1.0 subset (page 175) for details
- objects to which this property applies: <page-sequence>, <scaling-value-citation>

glyph-orientation-horizontal= (7.29.2; XSL; extended):

- Value: <angle> | inherit - Media: visual
- Initial: 0deg
- applied only to text written in a writing-mode with a left-to-right or right-to-left inline-progression-direction
- the only allowable angles are 0, 90, 180 and 270 degrees, counted clockwise
- "0deg" indicates the top of the glyph is towards the top of the reference-area
- object to which this property applies: <character>

glyph-orientation-vertical= (7.29.3; XSL; extended):

- Value: auto | <angle> | inherit - Media: visual
- Initial: auto
- applied only to text written in a writing-mode with a top-to-bottom or bottom-to-top inline-progression-direction
- the only allowable angles are 0, 90, 180 and 270 degrees, counted clockwise
- "0deg" indicates the top of the glyph is towards the top of the reference-area
- commonly used to differentiate between the preferred orientation of alphabetic text in vertically written Japanese documents ("auto") vs. the orientation of alphabetic text in western signage and advertising ("0deg")
- object to which this property applies: <character>

grouping-separator= (7.26.2; XSL; extended):

- Value: <character> - Media: all
- Initial: no separator

- see Format tokens from XSLT 1.0 subset (page 175) for details
- objects to which this property applies: <page-sequence>, <scaling-value-citation>

grouping-size= (7.26.3; XSL; extended):

- Value: <number> - Media: all
- Initial: no grouping
- see Format tokens from XSLT 1.0 subset (page 175) for details
- objects to which this property applies: <page-sequence>, <scaling-value-citation>

height= (7.15.6; CSS; basic):

- Value: <length> | <percentage> | auto | inherit - Percentages: see prose
- Initial: auto - Media: visual
- percentages are based on the explicit height of the containing block and are ignored if that height is not explicit
- negative values are illegal
- objects to which this property applies: <block-container>, <external-graphic>, <inline>, <inline-container>, <instream-foreign-object>, <table>, <table-caption>, <table-cell>, <table-row>

hyphenate= (7.10.4; XSL; extended):

- Value: false | true | inherit - Media: visual
- Initial: false
- specifies if hyphenation may or may not be used in the line-breaking algorithm for the text in the object
- objects to which this property applies: <block>, <character>

hyphenation-character= (7.10.5; XSL; extended):

- Value: <character> | inherit - Media: visual
- Initial: The Unicode hyphen character U+2010
- specifies the hyphenation character to be used when a hyphenation break occurs
- objects to which this property applies: <block>, <character>

hyphenation-keep= (7.16.1; XSL; extended):

- Value: auto | column | page | inherit - Media: visual
- Initial: auto
- specifies whether hyphenation can be performed on the last line that fits in a given reference-area
- object to which this property applies: <block>

hyphenation-ladder-count= (7.16.2; XSL; extended):

- Value: no-limit | <number> | inherit - Media: visual
- Initial: no-limit
- specifies a limit on the number of successive hyphenated line-areas the formatter may generate in a block-area
- object to which this property applies: <block>

hyphenation-push-character-count= (7.10.6; XSL; extended):

- Value: <number> | inherit - Media: visual
- Initial: 2

- specifies the minimum number of characters in a hyphenated word after the hyphenation character
- objects to which this property applies: <block>, <character>

hyphenation-remain-character-count= (7.10.7; XSL; extended):

- Value: <number> | inherit
- Media: visual
- Initial: 2
- specifies the minimum number of characters in a hyphenated word before the hyphenation character
- objects to which this property applies: <block>, <character>

id= (7.30.8; XSL; basic):

- Value: <id>
- Media: all
- Initial: see prose
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <block-container>, <character>, <external-graphic>, <float>, <flow>, <footnote>, <footnote-body>, <index-range-begin>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <list-item-body>, <list-item-label>, <multi-case>, <multi-property-set>, <multi-switch>, <multi-toggle>, <page-number>, <page-number-citation>, <page-number-citation-last>, <page-sequence>, <page-sequence-wrapper>, <root>, <scaling-value-citation>, <static-content>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <wrapper>

¶ index-class= (7.24.1; XSL; extended):

- Value: <string>
- Media: all
- Initial: empty string
- an annotation for index keys that distinguishes resolved index page numbers in a given index entry
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <block-container>, <character>, <external-graphic>, <float>, <flow>, <footnote>, <footnote-body>, <index-range-begin>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <list-item-body>, <list-item-label>, <multi-case>, <multi-property-set>, <multi-switch>, <multi-toggle>, <page-number>, <page-number-citation>, <page-number-citation-last>, <page-sequence>, <page-sequence-wrapper>, <root>, <scaling-value-citation>, <static-content>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <wrapper>

¶ index-key= (7.24.2; XSL; extended):

- Value: <string>
- Media: all
- Initial: none

- the key phrase used to distinguish index entries from each other
- there may be many uses of a given index key in order to create many page numbers in the one entry for that key value
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <block-container>, <character>, <external-graphic>, <float>, <flow>, <footnote>, <footnote-body>, <index-range-begin>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <list-item-body>, <list-item-label>, <multi-case>, <multi-property-set>, <multi-switch>, <multi-toggle>, <page-number>, <page-number-citation>, <page-number-citation-last>, <page-sequence>, <page-sequence-wrapper>, <root>, <scaling-value-citation>, <static-content>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <wrapper>

indicate-destination= (7.23.7; XSL; extended):

- Value: true | false
- Initial: false
- specifies that areas that belong to the link target when traversed should, in a system-dependent manner, be indicated
- object to which this property applies: <basic-link>
- Media: interactive

initial-page-number= (7.27.7; XSL; basic):

- Value: auto | auto-odd | auto-even | <number> | inherit
- Initial: auto
- object to which this property applies: <page-sequence>
- Media: visual

inline-progression-dimension= (7.15.7; CSS; basic):

- Value: auto | <length> | <percentage> | <length-range> | inherit
- Initial: auto
- a range allows the size to be adjusted by the formatter
- this does not apply when line-height= applies to the same dimension of the areas generated by this formatting object
- objects to which this property applies: <block-container>, <external-graphic>, <inline>, <inline-container>, <instream-foreign-object>, <table>, <table-caption>, <table-cell>
- Percentages: see prose
- Media: visual

internal-destination= (7.23.8; XSL; extended):

- Value: empty string | <idref>
- Initial: empty string
- at least one of external-destination= and internal-destination= properties should be assigned
- if both are assigned, the system may either report the error, or use internal-destination=
- objects to which this property applies: <basic-link>, <bookmark>
- Media: interactive

1 intrinsic-scale-value= (7.30.9; XSL; extended):

- Value: <percentage> | inherit
- Initial: 100%
- informs the processor of the scale the provided image is compared to the original image from which the provided image is obtained
- supplying this value then gives the processor the information to report scaling based on the original image and not the provided image
- object to which this property applies: <scaling-value-citation>
- Percentages: user defined
- Media: visual

intrusion-displace= (7.19.3; XSL; extended):

- Value: auto | none | line | indent | block | inherit
- Initial: auto
- this specifies the displacement strategy in the presence of intrusions (e.g. side floats)
- "none" allows any present intrusion to overlay the given line or block areas
- "block" reduces the inline progression dimension of the entire block that is in any way affected by the presence of an intrusion
- "line" reduces the inline progression dimension of only those line areas affected by the presence of an intrusion
- "indent" will act as "line" but will also preserve for each line any indentation relative to all other lines impacted by the given intrusion
- "auto" assumes "block" for reference-areas and "line" for other areas
- objects to which this property applies: <block>, <block-container>, <list-block>, <list-item>, <table>, <table-and-caption>, <table-caption>
- Media: visual

keep-together= (7.20.3; XSL; extended):

- Value: <keep> | inherit
- Initial: .within-line=auto, .within-column=auto, .within-page=auto
- objects to which this property applies: <basic-link>, <block>, <block-container>, <inline>, <inline-container>, <list-block>, <list-item>, <list-item-body>, <list-item-label>, <table>, <table-and-caption>, <table-caption>, <table-row>
- shorthand impacting on this property: page-break-inside=
- Media: visual

keep-with-next= (7.20.4; XSL; basic):

- Value: <keep> | inherit
- Initial: .within-line=auto, .within-column=auto, .within-page=auto
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-row>
- shorthands impacting on this property: page-break-after=, page-break-before=
- Media: visual

keep-with-previous= (7.20.5; XSL; basic):

- Media: visual

- Value: <keep> | inherit
- Initial: .within-line=auto, .within-column=auto, .within-page=auto
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-row>

language= (7.10.2; XSL; extended):

- Value: none | <language> | inherit
- Initial: none
- specifies the language used in language- and locale- coupled services
  - e.g. line-justification, line-breaking, hyphenation, etc.
- objects to which this property applies: <block>, <character>, <page-sequence>, <scaling-value-citation>
- shorthand impacting on this property: xml:lang=

- Media: visual

last-line-end-indent= (7.16.3; XSL; extended):

- Value: <length> | <percentage> | inherit
- Initial: Opt
- Percentages: refer to inline-progression-dimension of closest ancestor block-area that is not a line-area
- Media: visual
- positive values indent the end edge, negative values outdent the end edge
- object to which this property applies: <block>

leader-alignment= (7.22.1; XSL; extended):

- Value: none | reference-area | page | inherit
- Initial: none
- specifies whether all <leader> objects having identical content and property values shall have their patterns aligned with each other, with respect to their common reference-area or page
- "reference-area" specifies the pattern is aligned to the content-rectangle start-edge
- "page" specifies the pattern is aligned to the page's start-edge
- object to which this property applies: <leader>

- Media: visual

leader-length= (7.22.4; XSL; basic):

- Value: <length-range> | <percentage> | inherit
- Initial: leader-length.minimum=0pt, .optimum=12.0pt, .maximum=100%
- Percentages: refer to the inline-progression-dimension of content-rectangle of parent area
- Media: visual
- object to which this property applies: <leader>

leader-pattern= (7.22.2; XSL; basic):

- Value: space | rule | dots | use-content | inherit
- Initial: space
- Media: visual

- "use-content" specifies the children of the <leader> comprise the repeating pattern
- "space", "rule" and "dots" are available in all implementations
- an implementation may interpret "use-content" as "space"
- object to which this property applies: <leader>

leader-pattern-width= (7.22.3; XSL; extended):

- Value: use-font-metrics | <length> | <percentage> | inherit
- Initial: use-font-metrics
- Percentages: refer to the inline-progression-dimension of content-rectangle of parent area
- Media: visual
- object to which this property applies: <leader>

left= (7.6.5; CSS; extended):

- Value: <length> | <percentage> | auto | inherit
- Initial: auto
- Percentages: refer to width of containing block
- Media: visual
- specifies the distance between the margin edge and the containing block
- a value other than "auto" overrides a width= value of "auto"
- the end-indent= property is adjusted to correspond to any specified geometry of the content rectangle
- the percentage is ignored if the containing block width is not specified explicitly
- object to which this property applies: <block-container>

letter-spacing= (7.17.2; CSS; extended):

- Value: normal | <length> | <space> | inherit
- Initial: normal
- Media: visual
- specifies an amount to add (may be negative) to the built-in inter-character spacing
- specifying a value other than "normal" turns off the use of ligatures
- specifying a length turns off the modifications of inter-character spacing, but not the inter-word spacing, for justifying text
- objects to which this property applies: <bidirectional-override>, <character>, <initial-property-set>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>

letter-value= (7.26.4; XSL; basic):

- Value: auto | alphabetic | traditional
- Initial: auto
- Media: all
- see Format tokens from XSLT 1.0 subset (page 175) for details
- "auto" corresponds to an unspecified attribute in XSLT
- objects to which this property applies: <page-sequence>, <scaling-value-citation>

line-height= (7.16.4; CSS; basic):

- Value: normal | <length> | <number> | <percentage> | <space> | inherit
- Initial: normal
- Percentages: refer to the font size of the element itself
- Media: visual

- specifies the minimal height for a block-level construct of each generated inline box
- specifies the exact height for an inline-level construct
- the value "normal" is implementation specific, typically between 1.0 and 1.2
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <title>
- shorthand impacting on this property: font=

line-height-shift-adjustment= (7.16.5; XSL; extended):

- Value: consider-shifts | disregard-shifts | inherit - Media: visual
- Initial: consider-shifts
- specifies if the line-height of the line is adjusted for content that has a baseline shift
- a value of "disregard-shifts" will prevent superscript and subscript characters from disrupting the line-spacing
- see Chapter 4 Area and page basics (page 76) for more details on half-leading
- see Chapter 4 Area and page basics (page 76) for more details on superscripts and subscripts
- object to which this property applies: <block>

line-stacking-strategy= (7.16.6; XSL; basic):

- Value: line-height | font-height | max-height | inherit - Media: visual
- Initial: max-height
- specifies the strategy for positioning adjacent lines relative to each other
- see Chapter 4 Area and page basics (page 76) for details
- "line-height" may be interpreted as "max-height"
- object to which this property applies: <block>

linefeed-treatment= (7.16.7; XSL; extended):

- Value: ignore | preserve | treat-as-space | treat-as-zero-width-space | inherit - Media: visual
- Initial: treat-as-space
- object to which this property applies: <block>

margin= (7.31.14; CSS; shorthand):

- Value: margin-width{1,4} | inherit - Percentages: refer to width of containing block
- Initial: not defined for shorthand properties - Media: visual
- margin-width is one of <length>, <percentage> or "auto"
- up to four individual values for margins can be specified in this shorthand
- four values specifies each of the margins in the order: margin-top=, margin-right=, margin-bottom=, margin-left=
- three values specifies the margins in the order: margin-top=, margin-right=/margin-left=, margin-bottom=
- two values specifies the margins in the order: margin-top=/margin-bottom=, margin-right=/margin-left=
- one value specifies all the margins



margin-bottom= (7.11.2; CSS; basic):

- Value: margin-width | inherit
- Initial: 0pt
- Percentages: refer to width of containing block
- Media: visual
- margin-width is one of <length>, <percentage> or "auto"
- objects to which this property applies: <block>, <block-container>, <list-block>, <list-item>, <region-body>, <simple-page-master>, <table>, <table-and-caption>
- shorthand impacting on this property: margin=

margin-left= (7.11.3; CSS; basic):

- Value: margin-width | inherit
- Initial: 0pt
- Percentages: refer to width of containing block
- Media: visual
- margin-width is one of <length>, <percentage> or "auto"
- objects to which this property applies: <block>, <block-container>, <list-block>, <list-item>, <region-body>, <simple-page-master>, <table>, <table-and-caption>
- shorthand impacting on this property: margin=

margin-right= (7.11.4; CSS; basic):

- Value: margin-width | inherit
- Initial: 0pt
- Percentages: refer to width of containing block
- Media: visual
- margin-width is one of <length>, <percentage> or "auto"
- objects to which this property applies: <block>, <block-container>, <list-block>, <list-item>, <region-body>, <simple-page-master>, <table>, <table-and-caption>
- shorthand impacting on this property: margin=

margin-top= (7.11.1; CSS; basic):

- Value: margin-width | inherit
- Initial: 0pt
- Percentages: refer to width of containing block
- Media: visual
- margin-width is one of <length>, <percentage> or "auto"
- objects to which this property applies: <block>, <block-container>, <list-block>, <list-item>, <region-body>, <simple-page-master>, <table>, <table-and-caption>
- shorthand impacting on this property: margin=

marker-class-name= (7.25.1; XSL; extended):

- Value: <name>
- Initial: an empty name
- Media: paged
- identifies the <marker> as being in a group with others that have the same name
- object to which this property applies: <marker>

master-name= (7.27.8; XSL; basic):

- Media: visual

- Value: <name>
- Initial: an empty name
- objects to which this property applies: <page-sequence-master>, <simple-page-master>

master-reference= (7.27.9; XSL; basic):

- Value: <name>
- Initial: an empty name
- objects to which this property applies: <conditional-page-master-reference>, <page-sequence>, <repeatable-page-master-reference>, <single-page-master-reference>
- Media: visual

max-height= (7.15.8; CSS; complete):

- Value: <length> | <percentage> | none | inherit
- Initial: none
- Percentages: refer to height of containing block
- Media: visual
- percentages are based on the explicit height of the containing block and are ignored if that height is not explicit

max-width= (7.15.9; CSS; complete):

- Value: <length> | <percentage> | none | inherit
- Initial: none
- Percentages: refer to width of containing block
- Media: visual
- this property is mapped to either the inline-progression-dimension= or the block-progression-dimension=, based on the applicable values of the writing-mode= and reference-orientation= properties

maximum-repeats= (7.27.10; XSL; extended):

- Value: <number> | no-limit | inherit
- Initial: no-limit
- Media: visual
- objects to which this property applies: <repeatable-page-master-alternatives>, <repeatable-page-master-reference>

media-usage= (7.27.11; XSL; extended):

- Value: auto | paginate | bounded-in-one-dimension | unbounded
- Initial: auto
- Percentages: NA
- Media: visual
- specifies how the selected display medium is used to present the page(s) specified by the stylesheet
- "unbounded" generates only one page per <page-sequence> and requires that neither page-height= nor page-width= be specified
- "bounded-in-one-dimension" generates only one page per <page-sequence> and requires exactly one of either page-height= or page-width= to be specified on the first page master that is used
- object to which this property applies: <root>

1 merge-pages-across-index-key-references= (7.24.6; XSL; extended):

- Value: merge | leave-separate
- Initial: merge
- Media: all

- instructs the processor to throw away duplicate page number citations found from different key references, or leave them distinct in the massaged set of page numbers because the user needs to see duplicate references from different index keys
- object to which this property applies: `<index-page-citation-list>`

④ `merge-ranges-across-index-key-references=` (7.24.4; XSL; extended):

- Value: `merge` | `leave-separate` - Media: all
- Initial: `merge`
- instructs the processor to merge page ranges from different key references into single page ranges, rather than leaving them as distinct page ranges
- object to which this property applies: `<index-page-citation-list>`

④ `merge-sequential-page-numbers=` (7.24.5; XSL; extended):

- Value: `merge` | `leave-separate` - Media: all
- Initial: `merge`
- instructs the processor to change references to adjacent page numbers into a page range
- object to which this property applies: `<index-page-citation-list>`

`min-height=` (7.15.10; CSS; complete):

- Value: `<length>` | `<percentage>` | `inherit` - Percentages: refer to height of containing block
- Initial: `0pt` - Media: visual
- percentages are based on the explicit height of the containing block and are ignored if that height is not explicit

`min-width=` (7.15.11; CSS; complete):

- Value: `<length>` | `<percentage>` | `inherit` - Percentages: refer to width of containing block
- Initial: depends on UA - Media: visual
- percentages are based on the explicit height of the containing block and are ignored if that height is not explicit

`number-columns-repeated=` (7.28.12; XSL; basic):

- Value: `<number>` - Media: visual
- Initial: `1`
- specifies the number of consecutive columns (including the current column) for which the current column's specification applies
- object to which this property applies: `<table-column>`

`number-columns-spanned=` (7.28.13; XSL; basic):

- Value: `<number>` - Media: visual
- Initial: `1`
- specifies for `<table-cell>` the number of columns (including the current column) spanned by the one cell
- specifies for `<table-column>` the number of columns that may use properties from this column when using `from-table-column()`
- objects to which this property applies: `<table-cell>`, `<table-column>`

`number-rows-spanned=` (7.28.14; XSL; basic):

- Value: <number>
- Initial: 1
- object to which this property applies: <table-cell>
- Media: visual

odd-or-even= (7.27.12; XSL; extended):

- Value: odd | even | any | inherit
- Initial: any
- object to which this property applies: <conditional-page-master-reference>
- Media: visual

orphans= (7.20.6; CSS; basic):

- Value: <integer> | inherit
- Initial: 2
- specifies the minimum number of lines of a block that must be left at the bottom of a page
- object to which this property applies: <block>
- Media: visual

overflow= (7.21.2; CSS; basic):

- Value: visible | hidden | scroll | error-if-overflow | repeat | auto | inherit
- Initial: auto
- "visible" allows overflow content to be rendered outside the box
- "hidden" clips overflow content
- "scroll" provides a scrolling region for the entire content
- "error-if-overflow" is the same as "hidden" but also triggers an error
- objects to which this property applies: <block-container>, <external-graphic>, <inline-container>, <instream-foreign-object>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>
- Media: visual

padding= (7.31.15; CSS; shorthand):

- Value: padding-width{1,4} | inherit
- Initial: not defined for shorthand properties
- Percentages: refer to width of containing block
- Media: visual
- padding-width is one of <length> or <percentage>
- up to four individual values for padding values can be specified in this shorthand
- four values specifies each of the padding values in the order: padding-top=, padding-right=, padding-bottom=, padding-left=
- three values specifies the padding values in the order: padding-top=, padding-right=/padding-left=, padding-bottom=
- two values specifies the padding values in the order: padding-top=/padding-bottom=, padding-right=/padding-left=
- one value specifies all the padding values

padding-after= (7.8.32; CSS; basic):

- Value: padding-width | <length-conditional> | inherit
- Initial: Opt
- Percentages: refer to width of containing block
- Media: visual

- *padding-width* is one of <length> or <percentage>
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>

padding-before= (7.8.31; CSS; basic):

- Value: *padding-width* | <length-conditional> | inherit
- Initial: Opt
- Percentages: refer to width of containing block
- Media: visual
- *padding-width* is one of <length> or <percentage>
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>

padding-bottom= (7.8.36; CSS; basic):

- Value: *padding-width* | inherit
- Initial: Opt
- Percentages: refer to width of containing block
- Media: visual
- *padding-width* is one of <length> or <percentage>
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: padding=

padding-end= (7.8.34; CSS; basic):

- Value: *padding-width* | <length-conditional> | inherit
- Initial: Opt
- Percentages: refer to width of containing block
- Media: visual

- padding-width is one of <length> or <percentage>
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>

padding-left= (7.8.37; CSS; basic):

- Value: padding-width | inherit
- Initial: Opt
- Percentages: refer to width of containing block
- Media: visual

- padding-width is one of <length> or <percentage>
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: padding=

padding-right= (7.8.38; CSS; basic):

- Value: padding-width | inherit
- Initial: Opt
- Percentages: refer to width of containing block
- Media: visual

- padding-width is one of <length> or <percentage>
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: padding=

padding-start= (7.8.33; CSS; basic):

- Value: padding-width | <length-conditional> | inherit
- Initial: Opt
- Percentages: refer to width of containing block
- Media: visual

- padding-width is one of <length> or <percentage>
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>

padding-top= (7.8.35; CSS; basic):

- Value: padding-width | inherit
- Initial: 0pt
- Percentages: refer to width of containing block
- Media: visual

- padding-width is one of <length> or <percentage>
- objects to which this property applies: <basic-link>, <block>, <block-container>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: padding=

page-break-after= (7.31.16; CSS; shorthand):

- Value: auto | always | avoid | left | right | inherit
- Initial: auto
- Media: visual

Value	break-after=	keep-with-next=
"auto"	"auto"	"auto"
"always"	"page"	"auto"
"avoid"	"auto"	"always"
"left"	"even-page"	"auto"
"right"	"odd-page"	"auto"

page-break-before= (7.31.17; CSS; shorthand):

- Value: auto | always | avoid | left | right | inherit
- Initial: auto
- Media: visual

Value	break-before=	keep-with-previous=
"auto"	"auto"	"auto"
"always"	"page"	"auto"
"avoid"	"auto"	"always"
"left"	"even-page"	"auto"
"right"	"odd-page"	"auto"

page-break-inside= (7.31.18; CSS; shorthand):

- Value: avoid | auto | inherit
- Initial: auto
- Media: visual

- "auto" infers keep-together= of "auto"
- "avoid" infers keep-together= of "always"

❏ page-citation-strategy= (7.30.10; XSL; extended):

- Value: [ all | normal | non-blank | inherit - Media: visual
- Initial: all
- indicates which pages to include in the determination of the last page: all pages, only those pages that are have flow (i.e. are not out-of-line or blank), or only those pages that are not blank (i.e. include pages that have out-of-line content)
- object to which this property applies: <page-number-citation-last>

page-height= (7.27.13; XSL; basic):

- Value: auto | indefinite | <length> | inherit - Media: visual
- Initial: auto
- "auto" determines the value from the window (for continues media) or the size of the page
- "indefinite" determines the value from the laid-out content
  - overrides a value of "indefinite" for page-width= to be "auto"
- object to which this property applies: <simple-page-master>
- shorthand impacting on this property: size=

❏ page-number-treatment= (7.24.3; XSL; extended):

- Value: link | no-link - Media: interactive
- Initial: no-link
- in an index, indicate if the page numbers are hyperlinked to the page or not
- object to which this property applies: <index-key-reference>

page-position= (7.27.14; XSL; extended):

- Value: only | first | last | rest | any | inherit - Media: visual
- Initial: any
- object to which this property applies: <conditional-page-master-reference>

page-width= (7.27.15; XSL; basic):

- Value: auto | indefinite | <length> | inherit - Media: visual
- Initial: auto
- "auto" determines the value from the window (for continues media) or the size of the page
- "indefinite" determines the value from the laid-out content
  - overridden to be "auto" by a value of "indefinite" for page-height=
- object to which this property applies: <simple-page-master>
- shorthand impacting on this property: size=

pause= (7.31.19; CSS; shorthand):

- Value: [<time> | <percentage>]{1,2} | inherit - Percentages: see descriptions of 'pause-before' and 'pause-after'
- Initial: depends on user agent - Media: aural
- a shorthand specifying, in order, pause-before= and pause-after=
- specifying one value applies to both properties



pause-after= (7.7.5; CSS; basic):

- Value: <time> | <percentage> | inherit
- Initial: depends on user agent
- specifies the pause to be observed after speaking an object's content
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: pause=

pause-before= (7.7.6; CSS; basic):

- Value: <time> | <percentage> | inherit
- Initial: depends on user agent
- specifies the pause to be observed after speaking an object's content
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>
- shorthand impacting on this property: pause=

pitch= (7.7.7; CSS; basic):

- Value: <frequency> | x-low | low | medium | high | x-high | inherit
- Initial: medium
- specifies the average pitch (a frequency) of the speaking voice
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

pitch-range= (7.7.8; CSS; basic):

- Value: <number> | inherit
- Initial: 50
- Media: aural

- specifies variation in average pitch
- a highly animated voice displays a high pitch range
  - i.e. one that is heavily inflected
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

play-during= (7.7.9; CSS; basic):

- Value: <uri-specification> mix? repeat? | auto | none | inherit
- Initial: auto
- <uri-specification> specifies the sound played as a background while the object's content is spoken
- "mix" specifies the sound inherited from the parent object is mixed with the object's background <uri-specification>
- "repeat" specifies a short background to be repeated while the object is being spoken
- "auto" specifies the sound of the parent's background is to be continued
- "none" specifies silence for the background of the object being spoken
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

position= (7.31.20; CSS; shorthand):

- Value: static | relative | absolute | fixed | inherit
- Initial: static
- Media: visual

Value	relative-position=	absolute-position=
"static"	"static"	"auto"
"relative"	"relative"	"auto"
"absolute"	"static"	"absolute"
"fixed"	"static"	"fixed"

precedence= (7.27.16; XSL; extended):

- Value: true | false | inherit
- Initial: false
- Media: visual
- specifies whether the inline-progression-dimension of the region extends to content-rectangle of the page-reference-area ("true") or only to the edges incurred by the adjacent regions ("false")
- objects to which this property applies: <region-after>, <region-before>

provisional-distance-between-starts= (7.30.12; XSL; basic):

- Value: <length> | <percentage> | inherit
- Initial: 24.0pt
- Percentages: refer to inline-progression-dimension of closest ancestor block-area that is not a line-area
- Media: visual
- see Chapter 5 Generic body constructs (page 82) for a discussion of this property
- object to which this property applies: <list-block>

provisional-label-separation= (7.30.11; XSL; basic):

- Value: <length> | <percentage> | inherit
- Initial: 6.0pt
- Percentages: refer to inline-progression-dimension of closest ancestor block-area that is not a line-area
- Media: visual
- see Chapter 5 Generic body constructs (page 82) for a discussion of this property
- object to which this property applies: <list-block>

ref-id= (7.30.13; XSL; extended):

- Value: <idref> | inherit
- Initial: none, value required
- objects to which this property applies: <index-range-end>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>
- Media: all

① ref-index-key= (7.24.7; XSL; extended):

- Value: <string>
- Initial: none, value required
- used as the search string of index keys when collecting pages for an index entry
- object to which this property applies: <index-key-reference>
- Media: all

reference-orientation= (7.21.3; XSL; extended):

- Value: 0 | 90 | 180 | 270 | -90 | -180 | -270 | inherit
- Initial: 0
- Media: visual
- degree values are counter-clockwise from "0" degrees at the top
  - note this is a simple <integer> and not an <angle> data type value
- objects to which this property applies: <block-container>, <inline-container>, <page-sequence>, <region-after>, <region-before>, <region-body>, <region-end>, <region-start>, <simple-page-master>

region-name= (7.27.17; XSL; basic):

- Value: xsl-region-body | xsl-region-start | xsl-region-end | xsl-region-before | xsl-region-after | <name>
- Initial: see prose
- Media: visual
- objects to which this property applies: <region-after>, <region-before>, <region-body>, <region-end>, <region-start>

① region-name-reference= (7.27.21; XSL; extended):

- Value: <name>
- Initial: none, a value is required
- Media: all

- identifies a region by name in a flow map
- object to which this property applies: <region-name-specifier>

relative-align= (7.14.6; XSL; extended):

- Value: before | baseline | inherit - Media: visual
- Initial: before
- specifies the alignment, in the block-progression-direction, between two or more areas
- objects to which this property applies: <list-item>, <table-cell>

relative-position= (7.13.5; CSS; extended):

- Value: static | relative | inherit - Media: visual
- Initial: static
- "static" stacks the area normally
- "relative" positions an area as if it were stacked, but the area does not affect the position of any other area
  - any such area that breaks over a page boundary is clipped to the page and the remainder is discarded
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <character>, <external-graphic>, <initial-property-set>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>
- shorthand impacting on this property: position=

rendering-intent= (7.18.3; XSL; extended):

- Value: auto | perceptual | relative-colorimetric | saturation | absolute-colorimetric | inherit - Media: visual
- Initial: auto
- this property is applicable primarily to color-profiles corresponding to CMYK color spaces
- the different options cause different methods to be used for translating colors to the color gamut of the target rendering device
- object to which this property applies: <color-profile>

retrieve-boundary= (7.25.5; XSL; extended):

- Value: page | page-sequence | document - Media: paged
- Initial: page-sequence
- specifies how far "back" in the flow the formatter will look for a marker, starting with the current page
- object to which this property applies: <retrieve-marker>

❏ retrieve-boundary-within-table= (7.25.2; XSL; extended):

- Value: table | table-fragment | page - Media: paged
- Initial: table

- specifies how far "back" in the flow the formatter will look for a marker in a table, starting with the current table
- object to which this property applies: <retrieve-table-marker>

retrieve-class-name= (7.25.3; XSL; extended):

- Value: <name> - Media: paged
- Initial: an empty name
- constrains that the <marker> whose children are retrieved by the <retrieve-marker> must have a marker-class-name= property value that is the same as the value of this property
- objects to which this property applies: <retrieve-marker>, <retrieve-table-marker>

retrieve-position= (7.25.4; XSL; extended):

- Value: first-starting-within-page | first-including-carryover | last-starting-within-page | last-ending-within-page - Media: paged
- Initial: first-starting-within-page
- specifies the preference for which <marker> children shall be retrieved by a <retrieve-marker>
- object to which this property applies: <retrieve-marker>

① retrieve-position-within-table= (7.25.6; XSL; extended):

- Value: first-starting | first-including-carryover | last-starting | last-ending - Media: paged
- Initial: first-starting
- specifies the preference for which <marker> children shall be retrieved by a <retrieve-table-marker>
- object to which this property applies: <retrieve-table-marker>

richness= (7.7.10; CSS; basic):

- Value: <number> | inherit - Media: aural
- Initial: 50
- specifies the richness, or brightness, of the speaking voice
- a rich voice will "carry" in a large room, a smooth voice will not
  - the term "smooth" refers to how the wave form looks when drawn
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

right= (7.6.3; CSS; extended):

- Value: <length> | <percentage> | auto | inherit - Percentages: refer to width of containing block
- Initial: auto - Media: visual

- see `left=` for details
- object to which this property applies: `<block-container>`

`role=` (7.5.2; XSL; basic):

- Value: `<string>` | `<uri-specification>` | `none` | `inherit` - Media: all
- Initial: `none`
- provides a hint for alternate rendering agents (aural readers, etc.) as to the role of the XML element or elements that were used to construct this formatting object
- `<uri-specification>` specifies an RDF resource
- objects to which this property applies: `<basic-link>`, `<block>`, `<bookmark>`, `<bookmark-title>`, `<change-bar-begin>`, `<change-bar-end>`, `<external-graphic>`, `<footnote>`, `<footnote-body>`, `<initial-property-set>`, `<inline>`, `<instream-foreign-object>`, `<leader>`, `<list-block>`, `<list-item>`, `<list-item-body>`, `<list-item-label>`, `<multi-case>`, `<multi-properties>`, `<multi-switch>`, `<multi-toggle>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<root>`, `<scaling-value-citation>`, `<table>`, `<table-and-caption>`, `<table-body>`, `<table-caption>`, `<table-cell>`, `<table-footer>`, `<table-header>`, `<table-row>`, `<title>`

`rule-style=` (7.22.5; XSL; basic):

- Value: `none` | `dotted` | `dashed` | `solid` | `double` | `groove` | `ridge` | `inherit` - Media: visual
- Initial: `solid`
- "none" and "solid" are always supported
- other values may be interpreted as "solid"
- object to which this property applies: `<leader>`

`rule-thickness=` (7.22.6; XSL; basic):

- Value: `<length>` - Media: visual
- Initial: `1.0pt`
- this only applies if the `leader-pattern=` is "rule"
- object to which this property applies: `<leader>`

① `scale-option=` (7.30.14; XSL; extended):

- Value: `width` | `height` | `inherit` - Media: visual
- Initial: `width`
- when retrieving a scaling factor, this indicates which of the two is being retrieved
- object to which this property applies: `<scaling-value-citation>`

`scaling=` (7.15.12; XSL; extended):

- Value: `uniform` | `non-uniform` | `inherit` - Media: visual
- Initial: `uniform`
- specifies whether scaling needs to preserve the intrinsic aspect ratio
- objects to which this property applies: `<external-graphic>`, `<instream-foreign-object>`

`scaling-method=` (7.15.13; XSL; extended):

- Media: visual

- Value: auto | integer-pixels | resample-any-method  
| inherit
- Initial: auto
- indicates the preference in the scaling/sizing tradeoff to be used when formatting bitmapped graphics
- objects to which this property applies: <external-graphics>, <instream-foreign-object>

score-spaces= (7.30.15; XSL; extended):

- Value: true | false | inherit
- Initial: true
- specifies whether the text-decoration property shall be applied to spaces
- objects to which this property applies: <bidirectional-override>, <character>, <initial-property-set>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>

- Media: visual

script= (7.10.3; XSL; extended):

- Value: none | auto | <script> | inherit
- Initial: auto
- specifies the script used in language- and locale- coupled services
  - e.g. line-justification, line-breaking, hyphenation, etc.
- objects to which this property applies: <block>, <character>

- Media: visual

show-destination= (7.23.9; XSL; extended):

- Value: replace | new
- Initial: replace
- specifies whether the destination resource should replace the current document view or open a new document view
- object to which this property applies: <basic-link>

- Media: interactive

size= (7.31.21; CSS; shorthand):

- Value: <length>{1,2} | auto | landscape | portrait | inherit
- Initial: auto
- a shorthand specifying, in order, page-width= and page-height=
  - specifying one value applies to both properties

- Media: visual

source-document= (7.5.1; XSL; basic):

- Value: <uri-specification> [<uri-specification>]\*  
| none | inherit
- Initial: none

- Media: all

- provides pointers back to the original XML documents used to create the XSL-FO input
- W3C Accessibility guidelines strongly encourage the use of this property either on `<root>` or on the first formatting object from each source document
- objects to which this property applies: `<basic-link>`, `<block>`, `<bookmark>`, `<bookmark-title>`, `<change-bar-begin>`, `<change-bar-end>`, `<external-graphic>`, `<footnote>`, `<footnote-body>`, `<initial-property-set>`, `<inline>`, `<instream-foreign-object>`, `<leader>`, `<list-block>`, `<list-item>`, `<list-item-body>`, `<list-item-label>`, `<multi-case>`, `<multi-properties>`, `<multi-switch>`, `<multi-toggle>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<root>`, `<scaling-value-citation>`, `<table>`, `<table-and-caption>`, `<table-body>`, `<table-caption>`, `<table-cell>`, `<table-footer>`, `<table-header>`, `<table-row>`, `<title>`

space-after= (7.11.6; XSL; basic):

- Value: `<space>` | inherit
- Initial: `space.minimum=0pt`, `.optimum=0pt`, `.maximum=0pt`, `.conditionality=discard`, `.precedence=0`
- objects to which this property applies: `<block>`, `<block-container>`, `<list-block>`, `<list-item>`, `<region-body>`, `<simple-page-master>`, `<table>`, `<table-and-caption>`
- Percentages: N/A (Differs from margin-bottom in CSS)
- Media: visual

space-before= (7.11.5; XSL; basic):

- Value: `<space>` | inherit
- Initial: `space.minimum=0pt`, `.optimum=0pt`, `.maximum=0pt`, `.conditionality=discard`, `.precedence=0`
- objects to which this property applies: `<block>`, `<block-container>`, `<list-block>`, `<list-item>`, `<region-body>`, `<simple-page-master>`, `<table>`, `<table-and-caption>`
- Percentages: N/A (Differs from margin-top in CSS)
- Media: visual

space-end= (7.12.5; XSL; basic):

- Value: `<space>` | `<percentage>` | inherit
- Initial: `space.minimum=0pt`, `.optimum=0pt`, `.maximum=0pt`, `.conditionality=discard`, `.precedence=0`
- objects to which this property applies: `<basic-link>`, `<character>`, `<external-graphic>`, `<inline>`, `<inline-container>`, `<instream-foreign-object>`, `<leader>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`, `<title>`
- Percentages: refer to inline-progression-dimension of closest ancestor block-area that is not a line-area
- Media: visual

space-start= (7.12.6; XSL; basic):

- Value: `<space>` | `<percentage>` | inherit
- Initial: `space.minimum=0pt`, `.optimum=0pt`, `.maximum=0pt`, `.conditionality=discard`, `.precedence=0`
- objects to which this property applies: `<basic-link>`, `<character>`, `<external-graphic>`, `<inline>`, `<inline-container>`, `<instream-foreign-object>`, `<leader>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`, `<title>`
- Percentages: refer to inline-progression-dimension of closest ancestor block-area that is not a line-area
- Media: visual



- objects to which this property applies: <basic-link>, <character>, <external-graphic>, <inline>, <inline-container>, <instream-foreign-object>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <title>

span= (7.21.4; XSL; extended):

- Value: none | all | inherit
- Initial: none
- objects to which this property applies: <block>, <block-container>

speak= (7.7.11; CSS; basic):

- Value: normal | none | spell-out | inherit
- Initial: normal
- specifies the manner by which text will be rendered aurally
- "none" suppresses the aural rendering to zero time (unlike setting the volume to zero)
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

speak-header= (7.7.12; CSS; basic):

- Value: once | always | inherit
- Initial: once
- specifies whether table headers are spoken before every cell, or only before a cell when that cell is associated with a different header than the previous cell
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

speak-numeral= (7.7.13; CSS; basic):

- Value: digits | continuous | inherit
- Initial: continuous
- specifies how numerals are spoken
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

speak-punctuation= (7.7.14; CSS; basic):

- Value: code | none | inherit
- Initial: none
- specifies how punctuation is spoken
- note that "borge" is not an available option
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

- Media: aural

speech-rate= (7.7.15; CSS; basic):

- Value: <number> | x-slow | slow | medium | fast | x-fast | faster | slower | inherit
- Initial: medium
- <number> specifies the speaking rate in words per minute (varying somewhat by language)
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

- Media: aural

src= (7.30.16; XSL; basic):

- Value: <uri-specification> | inherit
- Initial: none, value required
- objects to which this property applies: <color-profile>, <external-graphic>

- Media: visual

start-indent= (7.11.7; XSL; basic):

- Value: <length> | <percentage> | inherit
- Initial: Opt
- Percentages: refer to inline-progression-dimension of containing reference-area
- Media: visual
- objects to which this property applies: <block>, <block-container>, <list-block>, <list-item>, <region-body>, <simple-page-master>, <table>, <table-and-caption>

starting-state= (7.23.10; XSL; extended):

- Value: show | hide
- Initial: show
- specifies the <multi-case> that can be initially displayed
- objects to which this property applies: <bookmark>, <multi-case>

- Media: interactive

starts-row= (7.28.15; XSL; extended):

- Value: true | false
- Initial: false
- object to which this property applies: <table-cell>
- Media: visual

stress= (7.7.16; CSS; basic):

- Value: <number> | inherit
- Initial: 50
- specifies the height of "local peaks" in the intonation contour of a voice
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>
- Media: aural

suppress-at-line-break= (7.17.3; XSL; extended):

- Value: auto | suppress | retain | inherit
- Initial: auto
- "auto" will suppress a U+0020 space character if it is first or last in a line
- "suppress" and "retain" will act accordingly with the given character
- object to which this property applies: <character>
- Media: visual

switch-to= (7.23.11; XSL; extended):

- Value: xsl-preceding | xsl-following | xsl-any | <name>[ <name>]\*
- Initial: xsl-any
- specifies which <multi-case> object this <multi-toggle> will switch to when evoked
- object to which this property applies: <multi-toggle>
- Media: interactive

table-layout= (7.28.16; CSS; extended):

- Value: auto | fixed | inherit
- Initial: auto
- object to which this property applies: <table>
- Media: visual

table-omit-footer-at-break= (7.28.17; XSL; extended):

- Value: true | false
- Initial: false
- specifies if a table whose last area is not at the end of an area produced by the table should end with the content of the <table-footer> formatting object or not
- object to which this property applies: <table>
- Media: visual

table-omit-header-at-break= (7.28.18; XSL; extended):

- Value: true | false
- Initial: false
- specifies if a table whose first area is not at the beginning of an area produced by the table should start with the content of the <table-header> formatting object or not
- object to which this property applies: <table>
- Media: visual

target-presentation-context= (7.23.12; XSL; extended):

- Value: use-target-processing-context | <uri-specification> - Media: interactive
- Initial: use-target-processing-context
- specifies the limited context in which the resource should be presented if the external destination is a resource of a processed structured media type for which a limited presentational context makes sense
- object to which this property applies: <basic-link>

target-processing-context= (7.23.13; XSL; extended):

- Value: document-root | <uri-specification> - Media: interactive
- Initial: document-root
- specifies the root of a virtual document that the processor preparing the new presentation should process if the external destination is a resource of a processed structured media type
- object to which this property applies: <basic-link>

target-stylesheet= (7.23.14; XSL; extended):

- Value: use-normal-style-sheet | <uri-specification> - Media: interactive
- Initial: use-normal-style-sheet
- specifies the stylesheet to be used for processing the target resource
- object to which this property applies: <basic-link>

text-align= (7.16.9; CSS; basic):

- Value: start | center | end | justify | inside | outside | left | right | <string> | inherit - Media: visual
- Initial: start
- <string> applies only to <table-cell> and specifies that which cells in a table column will align
- "left" and "right" are interpreted respectively as "start" and "end"
- objects to which this property applies: <block>, <external-graphic>, <instream-foreign-object>, <table-and-caption>

text-align-last= (7.16.10; XSL; extended):

- Value: relative | start | center | end | justify | inside | outside | left | right | inherit - Media: visual
- Initial: relative
- "left" and "right" are interpreted respectively as "start" and "end"
- object to which this property applies: <block>

text-altitude= (7.29.4; XSL; extended):

- Value: use-font-metrics | <length> | <percentage> | inherit - Percentages: refer to font's em-height
- Initial: use-font-metrics - Media: visual
- specifies the "height" to be used for the ascent above the dominant baseline
- objects to which this property applies: <block>, <character>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>

text-decoration= (7.17.4; CSS; extended):

- Value: none | [ [ underline | no-underline ] || [ overline | no-overline ] || [ line-through | no-line-through ] || [ blink | no-blink ] ] | inherit
- Initial: none
- this is not inherited but descendant boxes are formatted with the same decoration
- the color of the decorations remains the same even if descendent elements use different colors
- objects to which this property applies: <character>, <initial-property-set>, <inline>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>
- Media: visual

text-depth= (7.29.5; XSL; extended):

- Value: use-font-metrics | <length> | <percentage> | inherit
- Initial: use-font-metrics
- specifies the "depth" to be used for the descent below the dominant baseline
- objects to which this property applies: <block>, <character>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>
- Percentages: refer to font's em-height
- Media: visual

text-indent= (7.16.11; CSS; basic):

- Value: <length> | <percentage> | inherit
- Initial: 0pt
- Percentages: refer to width of containing block
- Media: visual
- positive values indent the start edge, negative values outdent the start edge with a hanging indent
- object to which this property applies: <block>

text-shadow= (7.17.5; CSS; extended):

- Value: none | [<color> || <length> <length> <length>? ,]\* [<color> || <length> <length> <length>?] | inherit
- Initial: none
- Media: visual
- specifies a comma-separated list of shadow effects to be applied to the text of the element
- a shadow effect is the color used as a basis the effect, the horizontal distance to the right (positive) or left (negative) of the text, the vertical distance below (positive) or above (negative) the text, and optionally a blur radius
- objects to which this property applies: <character>, <initial-property-set>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>

text-transform= (7.17.6; CSS; extended):

- Value: capitalize | uppercase | lowercase | none | inherit
- Initial: none
- Media: visual

- this is deprecated in XSL-FO due to "severe internationalization issues"
- objects to which this property applies: <character>, <initial-property-set>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>

top= (7.6.2; CSS; extended):

- Value: <length> | <percentage> | auto | inherit
- Initial: auto
- Percentages: refer to height of containing block
- Media: visual
- see bottom= for details
- object to which this property applies: <block-container>

treat-as-word-space= (7.17.7; XSL; extended):

- Value: auto | true | false | inherit
- Initial: auto
- Media: visual
- specifies if the character shall be treated as a word space ("true") or as a normal letter ("false")
- object to which this property applies: <character>

unicode-bidi= (7.29.6; CSS; extended):

- Value: normal | embed | bidi-override | inherit
- Initial: normal
- Media: visual
- specifies the opening of an additional level of embedding of characters for the bidirectional algorithm
- using "embed" provides for nesting directionality for blocks of Unicode characters separated by characters with weak or neutral directionality, without impacting on those characters that have a defined directionality
- using "bidi-override" ignores any defined directionality and imposes direction on the enclosed characters
- object to which this property applies: <bidi-override>

vertical-align= (7.31.22; CSS; shorthand):

- Value: baseline | middle | sub | super | text-top | text-bottom | <percentage> | <length> | top | bottom | inherit
  - Initial: baseline
  - Percentages: refer to the 'line-height' of the element itself
  - Media: visual
  - values specify the vertical positioning of inline-level constructs
  - Value alignment-baseline=alignment-adjust=baseline-shift=baseline-shift=baseline-shift=
- |               |                    |              |            |        |
|---------------|--------------------|--------------|------------|--------|
| "baseline"    | "baseline"         | "auto"       | "baseline" | "auto" |
| "top"         | "before-edge"      | "auto"       | "baseline" | "auto" |
| "text-top"    | "text-before-edge" | "auto"       | "baseline" | "auto" |
| "middle"      | "middle"           | "auto"       | "baseline" | "auto" |
| "bottom"      | "after-edge"       | "auto"       | "baseline" | "auto" |
| "text-bottom" | "text-after-edge"  | "auto"       | "baseline" | "auto" |
| "sub"         | "baseline"         | "auto"       | "sub"      | "auto" |
| "super"       | "baseline"         | "auto"       | "super"    | "auto" |
| <percentage>  | "baseline"         | <percentage> | "baseline" | "auto" |
| <length>      | "baseline"         | <length>     | "baseline" | "auto" |

visibility= (7.30.17; CSS; extended):

- Value: visible | hidden | collapse | inherit
- Media: visual
- Initial: visible
- specifies whether the boxes generated by an element are rendered even while affecting layout
- objects to which this property applies: <block>, <character>, <inline>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table-body>, <table-column>, <table-footer>, <table-header>, <table-row>, <title>

voice-family= (7.7.17; CSS; basic):

- Value: [[specific-voice | generic-voice ],]\* [specific-voice | generic-voice ] | inherit
- Media: aural
- Initial: depends on user agent
- generic-voice values are voice families from the set "male", "female" and "child"
- specific-voice values are specific instances that may be recognized by the rendering agent
  - e.g. "comedian", "romeo", "juliet", etc.
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

volume= (7.7.18; CSS; basic):

- Value: <number> | <percentage> | silent | x-soft | soft | medium | loud | x-loud | inherit
- Percentages: refer to inherited value
- Initial: medium
- Media: aural
- specify the median volume of the waveform
- objects to which this property applies: <basic-link>, <bidirectional-override>, <block>, <change-bar-begin>, <change-bar-end>, <character>, <external-graphic>, <initial-property-set>, <inline>, <instream-foreign-object>, <leader>, <list-block>, <list-item>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>, <table>, <table-and-caption>, <table-body>, <table-caption>, <table-cell>, <table-footer>, <table-header>, <table-row>, <title>

white-space= (7.31.23; CSS; shorthand):

- Value: normal | pre | nowrap | inherit
- Media: visual
- Initial: normal
- specifies a set of properties that would otherwise be individually specified
- Value
 

linefeed-treatment	white-space-collapse	white-space-treatment	wrap-option
"normal"	"treat-as-space"	"true"	"ignore-if-normal"
"pre"	"preserve"	"false"	"no-wrap"
"nowrap"	"treat-as-space"	"true"	"no-wrap"

white-space-collapse= (7.16.12; XSL; extended):

- Value: false | true | inherit
- Initial: true
- specifies that a space, tab or carriage return character is ignored if it follows another such character or immediately precedes a linefeed character
- object to which this property applies: <block>

white-space-treatment= (7.16.8; XSL; extended):

- Value: ignore | preserve | ignore-if-before-linefeed | ignore-if-after-linefeed | ignore-if-surrounding-linefeed | inherit
- Initial: ignore-if-surrounding-linefeed
- specifies the behavior of space, tab and carriage return characters
- object to which this property applies: <block>

widows= (7.20.7; CSS; basic):

- Value: <integer> | inherit
- Initial: 2
- specifies the minimum number of lines of a block that must be left at the top of a page
- object to which this property applies: <block>

width= (7.15.14; CSS; basic):

- Value: <length> | <percentage> | auto | inherit
- Initial: auto
- negative values are illegal
- does not apply to non-replaced inline-level elements
- objects to which this property applies: <block-container>, <external-graphic>, <inline>, <inline-container>, <instream-foreign-object>, <table>, <table-caption>, <table-cell>

word-spacing= (7.17.8; CSS; extended):

- Value: normal | <length> | <space> | inherit
- Initial: normal
- specifies inter-word spacing behavior in addition to the default between words
- negative values are allowed
- objects to which this property applies: <bidirectional-override>, <character>, <initial-property-set>, <leader>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>

wrap-option= (7.16.13; XSL; basic):

- Value: no-wrap | wrap | inherit
- Initial: wrap
- specifies how line-breaking of the content is to be handled
- "no-wrap" will cause lines longer than the width of the content-rectangle to be considered an overflow condition on the reference-area
- objects to which this property applies: <block>, <inline>, <page-number>, <page-number-citation>, <page-number-citation-last>, <scaling-value-citation>



writing-mode= (7.29.7; XSL; basic):

- Value: lr-tb | rl-tb | tb-rl | tb-lr | bt-lr | bt-rl | lr-bt |  
rl-bt | lr-alternating-rl-bt | lr-alternating-rl-tb | lr-  
inverting-rl-bt | lr-inverting-rl-tb | tb-lr-in-lr-pairs  
| lr | rl | tb | inherit
- Media: visual
- Initial: lr-tb
- "lr" is a shorthand for "lr-tb"
- "rl" is a shorthand for "rl-tb"
- "tb" is a shorthand for "tb-rl"
- objects to which this property applies: <block-container>, <inline-container>,  
<page-sequence>, <region-after>, <region-before>, <region-body>,  
<region-end>, <region-start>, <simple-page-master>, <table>

Ⓛ xml:lang= (7.31.24; XSL; shorthand):

- Value: language-country | inherit
- Media: visual
- Initial: not defined for shorthand properties
- country-language is a language and/or country value in conformance with RFC-3066
- recognized as a shorthand for country= and language=

z-index= (7.30.18; CSS; extended):

- Value: auto | <integer> | inherit
- Media: visual
- Initial: auto
- constructs with higher index numbers are "in front" of those with lower index numbers and show through the constructs that are "behind"
- objects to which this property applies: <block-container>, <change-bar-begin>

## Annex E - Sample tool information



- 
- Introduction - Sample questions for vendors

PREVIEW EXCERPT

# Sample questions for vendors

Annex E - Sample tool information



Answers to the following questions may prove useful when trying to better understand an XSL-FO product offering from a vendor. The specific questions are grouped under topical questions. This by no means makes up a complete list of questions as you may have your own criteria to add, nonetheless, they do cover aspects of XSLT that may impact on the stylesheets and transformation specifications you write.

- how is the product identified?
  - what is the name of the XSL-FO processor in product literature?
  - which version of XSL-FO is supported?
  - to which email address or URL are questions forwarded for more information in general?
  - to which email address or URL are questions forwarded for more information specific to the answers to these technical questions?
- what are the details of the implementation and invocation of the XSL-FO processor?
  - which hardware/operating system platforms support the processor?
  - which character sets are supported for the input file encoding?
  - what is the XSLT processor used?
    - can it be replaced with another XSLT processor?
  - what is the XML processor used?
    - can it be replaced with another XML processor?
    - does the XML processor support minimally declared internal declaration subsets with only attribute list declarations of ID-typed attributes?
    - does the XML processor support XML Inclusions (Xinclude)?
    - does the XML processor support catalogues for public identifiers?
    - does the XML processor validate the source file?
      - can this be turned on and off?
  - can the processor be embedded in other applications?
    - can the processor be configured as a servlet in a web server?
  - is the source code of the processor available?
    - in what language is the processor written?
  - for Windows-based environments:
    - can the processor be invoked from the MSDOS command-line box?
    - can the processor be invoked from a GUI interface?
    - what other methods of invocation can be triggered (DLL, RPC, etc.)?
    - can error messages be explicitly redirected to a file using an invocation parameter (since, for example, Windows-95 does not allow for redirection of the standard error port to a file)?

# Sample questions for vendors (cont.)

Annex E - Sample tool information



- 
- which features of XSL-FO are supported?
    - does the processor respect the inherent writing directions of Unicode characters and the Unicode bidirectional text algorithm?
    - which constructs are absent or only partially supported?
    - what graphics formats does the product support?
      - does the formatter support relative URI values for external graphic resources?
      - does the formatter require a protocol specification in the URI?
      - are vector-based formats rendered at device resolution?
      - example of photographic formats
        - JPEG (supports "lossy" compression)
      - examples of pixel formats
        - BMP (Windows)
        - PNG (see <http://www.libpng.org/> and <http://burnallgifs.org> for information)
        - GIF
      - examples of vector formats
        - SVG (Scalable Vector Graphics)
        - WMF (Windows Meta File)
        - EPS (Encapsulated PostScript)
        - CGM (Computer Graphics Metafile)
    - which color schemes are supported?
  - which features of EXSLFO are supported?
    - see <http://www.exslfo.org> for details on community-defined extensions to the XSL-FO vocabulary that may or may not be supported by vendors
  - how are particular features implemented?
    - at what conformance level is each feature implemented?
    - what is the thickness of each of the border width named values?
    - which writing directions are supported?
    - which languages are supported for hyphenation?
    - which fonts are supported and how are fonts supported?
    - which ligatures are recognized for adjacent characters?
    - which properties and states of <basic-link> are supported and how?
      - does the user interface have a "back" function to retrace steps?

## Sample questions for vendors (cont.)

Annex E - Sample tool information



- 
- what convenience features are implemented?
    - can the resulting file be fragmented for ease of management or transmission?
    - can the result be packaged in groups of different page sizes (to support fold-outs)?
    - can different hoppers be specified for the printer for different paper selections?
  - what extension formatting objects and properties are implemented?
    - are there any vendor-defined extensions to the vocabulary?
    - what namespace URI is used to identify the vocabulary?
  - what output formats are supported?
    - PDF?
      - are internal links supported?
      - are external links supported?
      - is there an extension for turning on the PDF security bit in the result?
      - is there an extension for defining the "General Info" fields for title, subject, author, etc.
    - PostScript?
    - Windows GDI?
    - PCL?
    - TeX?
    - RTF?
    - TIFF?
    - AFP?
    - other?
  - what instream foreign object vocabulary namespaces are supported?
    - SVG?
    - MathML?
    - other?

## Where To Go From Here?

Conclusion - Practical Formatting Using XSL-FO



---

The work on XSL and XSLT continues:

- all XSLT, XPath and XSL-FO are now full W3C Recommendations
- long list of future feature considerations already being examined for new releases of the technology
- new products are continually being announced
- feedback is necessary from users like you!
  - use the XSL mail lists to contribute:
    - <http://www.mulberrytech.com/xsl/xsl-list/>
    - <http://groups.yahoo.com/group/XSL-FO>
    - <http://lists.w3.org/Archives/Public/www-xsl-fo/>
  - contact the XSL editors with comments about the specification:
    - [xsl-editors@w3.org](mailto:xsl-editors@w3.org)

# Colophon

Conclusion - Practical Formatting Using XSL-FO



These materials were produced using structured information technologies as follows:

- authored source materials
  - content in numerous XML files maintained as external general entities for a complete prose book that can be made into a subset for training
    - specification of applicability of constructs for each configuration
      - 45- and 90-minute lecture, half-, full-, two- and three-day lecture and hands-on instruction, and book (prose) configurations
    - an XSLT transformation creates the subset of effective constructs from applying applicability to the complete file
    - content from other presentations/tutorials included semantically (not syntactically) during construct assembly
  - customized appearance engaged with marked sections and both parameter and general entities
    - different host company logos and venue and date marginalia
    - changing a single external parameter entity to a key file includes suite of files for given appearance
- accessible rendition in HTML
  - an XSLT stylesheet produces a collection of HTML files using Saxon for multiple file output
  - mono-spaced fonts and list-depth notation conventions assist the comprehension of the material when using screen-reader software
- printed handout deliverables
  - an XSLT stylesheet produces an instance of XSL formatting objects (XSL-FO) for rendering
  - XPDF <http://www.foolabs.com/xpdf> extracts raw text from PDF files for the back-of-the-book index methodology published as a free resource by Crane Softwrights Ltd.
  - XEP by RenderX <http://www.renderx.com> produces PostScript from XSL-FO
  - GhostScript <http://www.GhostScript.com> produces PDF from PostScript
  - the iText <http://itext.sf.net> PDF manipulation library for Java is used for page imposition by a custom Python <http://www.python.org> program running under the Jython <http://www.jython.org> environment

# Obtaining a copy of the comprehensive tutorial

Conclusion - Practical Formatting Using XSL-FO



This comprehensive tutorial on XSL-FO is available for subscription purchase and free preview download:

- "Practical Formatting Using XSL-FO (Extensible Stylesheet Language Formatting Objects)" Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
  - the free download preview excerpt of the publication indicates the number of pages for each topic
- the cost of purchase includes all future updates to the materials with email notification
  - the materials are updated after new releases of the W3C specifications
  - the materials are updated after incorporating comments gleaned during presentations and from feedback from customers
- available in PDF
  - formatted as 1-up or 2-up book pages per imaged page
  - dimensions in either US-letter or A4 page sizes
  - available as either single sided or double sided, full-page or half-page bound
- accessible rendition available for use with screen readers
- free preview download includes full text of first three chapters and two useful annexes
- site-wide and world-wide staff licenses (one-time fee) are available

See <http://www.CraneSoftwrights.com/links/trn-20080127.htm> for more details.

## Feedback

- the unorthodox style has been well-accepted by customers as an efficient learning presentation
- feedback from customers is important to improve or repair the content for future editions
- please send suggestions or comments (positive or negative) to [info@CraneSoftwrights.com](mailto:info@CraneSoftwrights.com)

## US Government employee purchase

- US Government employees (not contractors) are entitled to obtain their personal prepaid copies at no charge from a government intranet location
- visit the Crane web site for details





# Practical Formatting Using XSL-FO (Extensible Stylesheet Language Formatting Objects)

Crane Softwrights Ltd.  
<http://www.CraneSoftwrights.com>

# Practical Formatting Using XSL-FO

Table of contents

Indexed by slide number



---

001	[Prelude ] Practical Formatting Using XSL-FO (Prelude) (002)
003	Practical Formatting Using XSL-FO
004	[Introduction -1-1] Paginating structured information (005)
006	[1] Introducing XSL-FO
007	[Introduction 1-1-1] Contrasting browsed vs. paginated presentations (008) (009) (010) (011) (012) (013)
014	[1-1-1-1] Fine-grained control of layout nuances
015	[1-2-1-1] The many existing deployments of XSL-FO
016	[2] The context of XSL-FO
017	[Introduction 2-1-1] Overview
018	[2-1-1-1] Extensible Markup Language (XML) (019)
020	[2-1-2-1] XML information links
021	[2-1-3-1] Document Style Semantics and Specification Language (DSSSL)
022	[2-1-4-1] Cascading Stylesheets (CSS)
023	[2-1-5-1] Styling structured information
024	[2-1-6-1] Extensible Stylesheet Language Transformations (XSLT) (025)
026	[2-1-7-1] Extensible Stylesheet Language (XSL/XSL-FO)
027	[2-1-8-1] Styling semantics and vocabularies
028	[2-1-9-1] Outboard XSLT and XSL-FO processes
029	[2-1-10-1] Transforming and rendering XML information using XSLT and XSL-FO
030	[2-1-11-1] Using XSL-FO as an intermediate form
031	[2-1-12-1] Generating XSL-FO instances
032	[2-1-13-1] Using XSL-FO on a server (033) (034)
035	[2-1-14-1] Historical development of the XSL and XSLT Recommendations
036	[2-1-15-1] XSL information links (037) (038)
039	[2-2-1-1] Hello world example (040)
041	[2-2-2-1] A detailed example of flowed content
042	[2-2-3-1] Training material example (043)
044	[3] Basic concepts of XSL-FO
045	[Introduction 3-1-1] Essential concepts and terminology (046)
047	[3-1-1-1] Layout-based vs. content-based formatting
048	[3-1-2-1] Formatting vs. Rendering (049)
050	[3-2-1-1] Processing model of formatting (051) (052)
053	[3-3-1-1] Vocabulary structure (054)
055	[3-3-2-1] Direct vs. constraint property specification
056	[3-3-3-1] Property value expressions
057	[3-3-4-1] Inherited properties
058	[3-3-5-1] Shorthand properties
059	[3-3-6-1] Conformance
060	[3-3-7-1] Top-level formatting objects
061	[3-3-8-1] <root> Object (062)
063	[3-3-9-1] <layout-master-set> Object (064)

065	[3-3-10-1]	<page-sequence> Object	(066)
067	[3-3-11-1]	<page-sequence-wrapper> Object	
068	[3-3-12-1]	<flow> Object	(069)
070	[3-4-1-1]	Groupings of formatting objects for flow	
071	[3-4-2-1]	Block-level objects	
072	[3-4-3-1]	Inline-level objects	
073	[3-4-4-1]	Neutral objects	
074	[3-4-5-1]	Out-of-line objects	
075	[3-4-6-1]	Out-of-line inline-level objects	
076	[4]	Area and page basics	
077	[Introduction 4-1-1]	Area and page overview	(078)
079	[Introduction 4-2-1]	Page geometry	(080)
081	[Introduction 4-3-1]	Area and page constructs	
082	[5]	Generic body constructs	
083	[Introduction 5-1-1]	Often-used formatting constructs	(084) (085) (086) (087)
088	[6]	Tables	
089	[Introduction 6-1-1]	Tabular presentation of information	(090) (091) (092) (093)
094	[7]	Floats, footnotes and containers	
095	[Introduction 7-1-1]	Out-of-line publishing constructs	(096) (097) (098)
099	[8]	Flows, static content and page geometry sequencing	
100	[Introduction 8-1-1]	Extended page model for pagination	(101) (102) (103) (104) (105)
106	[8-1-1-1]	Changing the page geometry based on authored content	(107) (108) (109) (110)
111	[9]	Bookmarks and indexes	
112	[Introduction 9-1-1]	Additional navigation aids	(113)
114	[10]	Breaks, keeps, spacing, borders and backgrounds	
115	[Introduction 10-1-1]	Spacing and arrangement constraint definition	(116)
117	[11]	Supplemental objects	
118	[Introduction 11-1-1]	Specialty constructs	(119)
120	[12]	Interactive objects	
121	[Introduction 12-1-1]	Dynamic and interactive object rendering	(122)
123	[13]	Where XSL-FO 1 falls short	
124	[Introduction 13-1-1]	What is missing in XSL-FO 1?	
125	[A]	Using XSLT with XSL-FO	
126	[Introduction A-1-1]	Using XSLT with XSL-FO	
127	[A-1-1-1]	<xsl:attribute-set> instruction	
128	[A-1-2-1]	Simpler list and footnote structures in XSL-FO	
129	[A-2-1-1]	Common errors writing expressions	(130)
131	[B]	XSL-FO expressions	
132	[Introduction B-1-1]	Expressions in XSL-FO	
133	[B-1-1-1]	Production summary	
134	[B-2-1-1]	Function summary	
135	[B-2-2-1]	Function groupings	
136	[B-2-3-1]	Functions summarized by name	
137	[C]	XSL-FO object summary	
138	[Introduction C-1-1]	Objects	
139	[C-1-1-1]	Objects summarized by name	

140	[C-1-2-1]	Prototypical hierarchy
141	[C-1-3-1]	Objects summarized by type
142	[D]	XSL-FO property summaries
143	[Introduction D-1-1]	Properties
144	[D-1-1-1]	Common Absolute Position Properties
145	[D-1-2-1]	Common Accessibility Properties
146	[D-1-3-1]	Common Aural Properties
147	[D-1-4-1]	Common Border, Padding, and Background Properties
148	[D-1-5-1]	Common Font Properties
149	[D-1-6-1]	Common Hyphenation Properties
150	[D-1-7-1]	Common Margin Properties-Block
151	[D-1-8-1]	Common Margin Properties-Inline
152	[D-1-9-1]	Common Relative Position Properties
153	[D-2-1-1]	Property data types
154	[D-2-2-1]	Recommendation bibliography
155	[D-2-3-1]	Format tokens from XSLT 1.0 subset
156	[D-3-1-1]	Inherited properties
157	[D-3-2-1]	Shorthand properties
158	[D-3-3-1]	Property summary
159	[E]	Sample tool information
160	[Introduction E-1-1]	Sample questions for vendors (161) (162)
163	[Conclusion -1-1]	Where To Go From Here?
164	[Conclusion -2-1]	Colophon
165	[Conclusion -3-1]	Obtaining a copy of the comprehensive tutorial
166	[Postlude ]	Practical Formatting Using XSL-FO (Postlude)



# Practical Formatting Using XSL-FO

Index

## A

A4 page dimensions 39  
 abs() function **138**  
   in function summaries 136  
 absolute-position= property **181**  
   in set of common properties 158  
   referenced 224  
 active-state= property **181**  
 Adobe Acrobat 40  
 alignment  
   blocks 83  
 alignment-adjust= property **181**  
   referenced 182, 236  
 alignment-baseline= property **182**  
   referenced 182, 236  
 allowed-height-scale= property **182**  
   in inherited property summary 176  
 allowed-width-scale= property **182**  
   in inherited property summary 176  
 alphabetic numbering 175  
 <angle> data type **169**  
   in property definition 183, 202, 207  
   referenced 2, 225  
 Antenna House XSL Formatter 40, 54  
 appearance 48  
 area  
   tree/model 52-53, 77-78  
 aural media 26, 29, 59  
 authoring 9  
 auto-restore= property **183**  
   in inherited property summary 176  
 azimuth= property **183**  
   in inherited property summary 176  
   in set of common properties 160

## B

background= property **183**  
   in object definition 161  
   in property definition 184, 186  
   in shorthand definition 179  
 background-attachment= property **183**  
   in set of common properties 161

background-color= property **184**  
   in set of common properties 161  
   referenced 184  
 background-image= property **184**  
   in set of common properties 161  
 background-position= property **184**  
   in object definition 161  
   in property definition 185  
   in shorthand definition 179  
 background-position-horizontal= prop-  
   erty **185**  
   in set of common properties 161  
   referenced 185  
 background-position-vertical= property  
   **185**  
   in set of common properties 161  
   referenced 185  
 background-repeat= property **185**  
   in set of common properties 161  
   referenced 184  
 baseline-shift= property **186**  
   referenced 2, 182, 236  
 <basic-link> object **87**  
   in object summaries 142, 153  
   in property definition 182-196, 201-203,  
   209-212, 214, 219-221, 223-224, 226-234,  
   237  
   referenced 2, 72, 87, 151, 159-161, 166-167,  
   181, 242  
 before-float-reference-area 97, 115  
 <bidirectional-override> object **119**  
   in object summaries 142, 154  
   in property definition 183, 199, 201-202,  
   205-207, 209-210, 213-214, 223-224,  
   226-227, 229, 231-233, 236-238  
   referenced 72, 119, 151, 160, 163, 167, 201  
 bilingual presentation 84, 89  
 blank page 60, 102  
 blank-or-not-blank= property **186**  
 %block; entity object **71**  
   in parent content model 68, 142-146  
   referenced 73-74

- `%inline;` entity object **72**
  - in parent content model 142-144, 146
  - referenced 73-75
- `<block>` object **81**
  - in object summaries 142, 153
  - in property definition 183-203, 205-212, 214-215, 218-221, 223-224, 226-235, 237-238
  - referenced 70-71, 81, 151, 159-161, 163-165, 167, 198
- `<block-container>` object **98**
  - in object summaries 142, 153
  - in property definition 181, 184-199, 202-203, 208-213, 215, 218-221, 225, 228, 230-232, 236, 238-239
  - referenced 71, 96, 98, 149, 158, 161, 165
- `block-progression-dimension=` property **186**
  - referenced 216
- `block-progression-direction` 115
- `body-start()` function **138**
  - in function summaries 136
- `<bookmark>` object **113**
  - in object summaries 142, 154
  - in parent content model 142
  - in property definition 203, 210, 228, 230, 232
  - referenced 113, 149, 159
- `<bookmark-title>` object **113**
  - in object summaries 142, 154
  - in parent content model 142
  - in property definition 199, 206-207, 228, 230
  - referenced 113, 149, 159
- `<bookmark-tree>` object **113**
  - in object summaries 142, 154
  - in parent content model 61, 145
  - referenced 113, 149
- `border=` property **186**
  - in object definition 161
  - in property definition 187-188, 190, 194
  - in shorthand definition 179
  - referenced 58
- `border-after-color=` property **186**
  - in set of common properties 161
- `border-after-precedence=` property **187**
  - referenced 188, 191, 194
- `border-after-style=` property **187**
  - in set of common properties 161
- `border-after-width=` property **187**
  - in set of common properties 161
- `border-before-color=` property **188**
  - in set of common properties 161
- `border-before-precedence=` property **188**
  - in set of common properties 161
- `border-before-style=` property **188**
  - in set of common properties 161
- `border-before-width=` property **188**
  - in set of common properties 161
- `border-bottom=` property **189**
  - in object definition 162
  - in property definition 189-190
  - in shorthand definition 179
  - referenced 58
- `border-bottom-color=` property **189**
  - in set of common properties 161
  - referenced 58
- `border-bottom-style=` property **189**
  - in set of common properties 161
- `border-bottom-width=` property **189**
  - in set of common properties 161
- `border-collapse=` property **190**
  - in inherited property summary 176
  - referenced 187, 193, 203
- `border-color=` property **190**
  - in object definition 162
  - in property definition 187-190, 192-195
  - in shorthand definition 179
  - referenced 58
- `border-end-color=` property **190**
  - in set of common properties 161
- `border-end-precedence=` property **190**
  - in set of common properties 161
- `border-end-style=` property **191**
  - in set of common properties 161
- `border-end-width=` property **191**
  - in set of common properties 161
- `border-left=` property **191**
  - in object definition 162
  - in property definition 192
  - in shorthand definition 179
- `border-left-color=` property **191**
  - in set of common properties 161

- ul style="list-style-type: none; padding-left: 0;">
- border-left-style= property **192**
  - in set of common properties 161
- border-left-width= property **192**
  - in set of common properties 161
- border-rectangle 77
- border-right= property **192**
  - in object definition 161
  - in property definition 193
  - in shorthand definition 179
- border-right-color= property **192**
  - in set of common properties 161
- border-right-style= property **193**
  - in set of common properties 161
- border-right-width= property **193**
  - in set of common properties 161
- border-separation= property **193**
  - in inherited property summary 176
  - referenced 194
- border-spacing= property **193**
  - in inherited property summary 176
  - in property definition 193
  - in shorthand definition 179
- border-start-color= property **194**
  - in set of common properties 161
- border-start-precedence= property **194**
- border-start-style= property **194**
  - in set of common properties 161
- border-start-width= property **194**
  - in set of common properties 161
- border-style= property **195**
  - in object definition 161
  - in property definition 187-189, 191-194, 196
  - in shorthand definition 179
  - referenced 58
- border-top= property **195**
  - in object definition 161
  - in property definition 195-196
  - in shorthand definition 179
- border-top-color= property **195**
  - in set of common properties 161
- border-top-style= property **195**
  - in set of common properties 161
- border-top-width= property **196**
  - in set of common properties 161
- border-width= property **196**
  - in object definition 162
  - in property definition 188-193, 195-196
  - in shorthand definition 179
- bottom= property **196**
  - in set of common properties 158
  - referenced 236
- break-after= property **196**
  - referenced 60, 221
- break-before= property **197**
  - referenced 60, 221
- browser windows 7, 13, 40
- C**
- caption-side= property **197**
  - in inherited property summary 176
- Cascading Stylesheets (CSS) 17, 19, 22, 23, 26, 45
- case-name= property **197**
- case-title= property **197**
- ceiling() function **138**
  - in function summaries 136
- change pages 47
- <change-bar-begin> object **119**
  - in object summaries 142, 155
  - in property definition 183, 197-198, 201-202, 223-224, 227-228, 230-233, 237, 239
  - referenced 73, 119, 151, 159-160
- change-bar-class= property **197**
- change-bar-color= property **197**
  - in inherited property summary 176
- <change-bar-end> object **119**
  - in object summaries 142, 155
  - in property definition 183, 197, 201-202, 223-224, 227-228, 230-233, 237
  - referenced 73, 119, 151, 159-160
- change-bar-offset= property **197**
  - in inherited property summary 176
- change-bar-placement= property **198**
  - in inherited property summary 176
- change-bar-style= property **198**
  - in inherited property summary 176
- change-bar-width= property **198**
  - in inherited property summary 176
- <character> data type **169**
  - in property definition 198, 207-208

- <character> object **119**
  - in object summaries 142, 154
  - in property definition 182-196, 198-202, 205-214, 219-221, 223-224, 226-227, 229-238
  - referenced 72, 119, 151, 160-161, 163-164, 166-167
- character= property **198**
- clear= property **198**
- clip= property **198**
- colophon 245
- <color> data type **169**
  - in property definition 184, 186-192, 194-195, 197, 199, 235
- color= property **199**
  - in inherited property summary 176
  - referenced 55
- <color-profile> object **119**
  - in object summaries 142, 153
  - in parent content model 142
  - in property definition 199, 226, 232
  - referenced 119, 149
- color-profile-name= property **199**
- column-count= property **199**
- column-gap= property **199**
- column-number= property **199**
  - referenced 199
- column-width= property **199**
- columns (page)
  - single, see normal-flow-reference-area
- columns (table)
  - properties 92
  - spanned 91-92
  - width 89-90
- "Common Absolute Position Properties" property set **158**
- "Common Accessibility Properties" property set **159**
  - in object definition 61
- "Common Aural Properties" property set **160**
- "Common Border, Padding, and Background Properties" property set **161-162**
- "Common Font Properties" property set **163**
- "Common Hyphenation Properties" property set **164**
- "Common Margin Properties-Block" property set **165**
- "Common Margin Properties-Inline" property set **166**
- "Common Relative Position Properties" property set **167**
- compound properties 53, 58
- <conditional-page-master-reference> object **105**
  - in object summaries 142, 153
  - in parent content model 145
  - in property definition 186, 216, 218, 222
  - referenced 105, 149
- conditionality 53, 116
- conformance 59
- container objects
  - block containers 95-98
- content-based formatting 45, 47
- content-height= property **199**
- content-rectangle 77
- content-type= property **200**
- content-width= property **200**
- <country> data type **169**
  - in property definition 200
- country= property **200**
  - in inherited property summary 176
  - in object definition 65
  - in set of common properties 164
  - referenced 239
- cue= property **200**
  - in object definition 160
  - in property definition 201
  - in shorthand definition 179
- cue-after= property **200**
  - in set of common properties 160
- cue-before= property **201**
  - in set of common properties 160
- D**
- data types 56, 168-171
- <declarations> object **119**
  - in object summaries 142, 153
  - in parent content model 61, 145
  - referenced 51, 119, 149
- destination-placement-offset= property **201**
- device independence 26



dictionary headers 101  
 direction= property **201**  
   in inherited property summary 176  
 discarding space  
   areas, see conditionality  
 display-align= property **201**  
   in inherited property summary 176  
 Document Object Model (DOM) 24  
 Document Style Semantics and Specification  
   Language (DSSSL) 17, 21, 26, 45  
 Document Type Definition (DTD) 27  
 dominant-baseline= property **202**  
   referenced 236  
 dot leader 83  
 dynamic content 80

## E

elevation= property **202**  
   in inherited property summary 176  
   in set of common properties 160  
 empty page, see blank page  
 empty-cells= property **202**  
   in inherited property summary 176  
 end-indent= property **203**  
   in inherited property summary 176  
   in set of common properties 165  
   referenced 213  
 endnotes 95  
 ends-row= property **203**  
 Examples ZIP file 5  
 expressions 56  
   errors writing 129-130  
   productions 133-134  
 Extensible Markup Language (XML) 17,  
   18-19  
 Extensible Stylesheet Language (XSL) 17  
 Extensible Stylesheet Language Formatting  
   Objects (XSL-FO) **13**, 17, 23, 26, 29, 30  
   vocabulary 13, 53-54  
 Extensible Stylesheet Language Transforma-  
   tions (XSLT) 11, 17, 23, 24-25, 106, 109,  
   126-130  
 extent= property **203**  
 external-destination= property **203**  
   referenced 203, 210

<external-graphic> object **87**  
   in object summaries 142, 154  
   in property definition 182-196, 199-202,  
   208-212, 214, 218-221, 223-224, 226-234,  
   237-238  
   referenced 72, 87, 151, 159-161, 166-167

## F

fallback processing 59  
 <family-name> data type **169**  
   in property definition 204  
 <float> object **98**  
   in object summaries 142, 155  
   in property definition 198, 203, 209-210  
   referenced 74, 96, 98, 150, 203  
 float= property **203**  
 floats 95-98  
 floor() function **138**  
   in function summaries 136  
 <flow> object **68**  
   in object summaries 142, 153  
   in parent content model 65, 145  
   in property definition 204, 209-210  
   referenced ?, 46, 60, 71-75, 149  
 flow semantics 26, 48  
 <flow-assignment> object **105**  
   in object summaries 142, 153  
   in parent content model 143  
   referenced 105, 149  
 <flow-map> object **105**  
   in object summaries 143, 153  
   in parent content model 63, 144  
   in property definition 204  
   referenced 105, 149  
 flow-map-name= property **203**  
 flow-map-reference= property **204**  
   in object definition 65  
 flow-name= property **204**  
   in object definition 68  
   referenced 68  
 flow-name-reference= property **204**  
 <flow-name-specifier> object **105**  
   in object summaries 143, 153  
   in parent content model 143  
   in property definition 204  
   referenced 105, 149

- `<flow-source-list>` object **105**
  - in object summaries 143, 153
  - in parent content model 142
  - referenced 105, 149
- `<flow-target-list>` object **105**
  - in object summaries 143, 153
  - in parent content model 142
  - referenced 105, 149
- `fo2html.xsl` 30
- `<folio-prefix>` object **104**
  - in object summaries 143, 154
  - in parent content model 65, 145
  - referenced 104, 149
- `<folio-suffix>` object **104**
  - in object summaries 143, 154
  - in parent content model 65, 145
  - referenced 104, 149
- `font=` property **204**
  - in inherited property summary 176
  - in object definition 163
  - in property definition 205-207, 214
  - in shorthand definition 179
  - referenced 58
- font size 51
- `font-family=` property **204**
  - in inherited property summary 176
  - in set of common properties 163
  - referenced 58, 205
- `font-selection-strategy=` property **205**
  - in inherited property summary 176
  - in set of common properties 163
  - referenced 205
- `font-size=` property **205**
  - in inherited property summary 176
  - in set of common properties 163
  - referenced 58, 205
- `font-size-adjust=` property **205**
  - in inherited property summary 176
  - in set of common properties 163
- `font-stretch=` property **206**
  - in inherited property summary 176
  - in set of common properties 163
  - referenced 205
- `font-style=` property **206**
  - in inherited property summary 176
  - in set of common properties 163
  - referenced 58, 205
- `font-variant=` property **206**
  - in inherited property summary 176
  - in set of common properties 163
  - referenced 58, 205
- `font-weight=` property **206**
  - in inherited property summary 176
  - in set of common properties 163
  - referenced 58, 205
- footers
  - page 101
  - table 92
- `<footnote>` object **98**
  - in object summaries 143, 155
  - in property definition 209-210, 228, 230
  - referenced 75, 96, 98, 151, 159
- `<footnote-body>` object **98**
  - in object summaries 143, 155
  - in parent content model 143
  - in property definition 209-210, 228, 230
  - referenced 98, 151, 159
- footnote-reference-area 97, 115
- footnotes 95-98
  - labeling 95, 128
- `force-page-count=` property **207**
  - in object definition 65
- `format=` property **207**
  - in object definition 65
  - referenced 65
- format tokens 175
- formatting 23, 45, 48-49
  - formatting semantics 26
  - page numbers 60, 175
- Formatting Object Tree 51
- formatting objects
  - block-level 70, 71
  - inline-level 70, 72
  - neutral 70, 73
  - out-of-line 70, 74, 95-98
  - out-of-line inline 70, 75, 95-98
  - summary list 142-146
- `<frequency>` data type **169**
  - in property definition 223

- from-nearest-specified-value() function **138**
  - in function summaries 136
- from-page-master-region() function **138**
  - in function summaries 136
- from-parent() function **138**
  - in function summaries 136
  - referenced 56
- from-table-column() function **138**
  - in function summaries 136
  - referenced 57, 217
- functions 57
  - library 56
  - summary list 138-139
- G**
  - general purpose XML transformations 24
  - glyph-orientation-horizontal= property **207**
    - in inherited property summary 176
  - glyph-orientation-vertical= property **207**
    - in inherited property summary 176
  - graphics 86
  - grouping-separator= property **207**
    - in object definition 65
  - grouping-size= property **208**
    - in object definition 65
- H**
  - headers
    - page 101
    - table 92
  - height= property **208**
    - referenced 196
  - hyperlink 8, 10-11, 83, 86
  - hyperlinks to outside information 20, 36-38
  - Hypertext Markup Language (HTML) 7, 9, 11, 13, 17, 19, 22, 30, 32-34, 89, 121
  - hyphenate= property **208**
    - in inherited property summary 176
    - in set of common properties 164
  - hyphenation-character= property **208**
    - in inherited property summary 176
    - in set of common properties 164
  - hyphenation-keep= property **208**
    - in inherited property summary 176
  - hyphenation-ladder-count= property **208**
    - in inherited property summary 176
  - hyphenation-push-character-count= property **208**
    - in inherited property summary 176
    - in set of common properties 164
  - hyphenation-remain-character-count= property **209**
    - in inherited property summary 176
    - in set of common properties 164
- I**
  - <id> data type **169**
    - in property definition 209
  - id= property **209**
    - in object definition 61, 65, 67-68
    - referenced 54
  - <idref> data type **169**
    - in property definition 210, 225
  - impartation semantics 48, 52
  - index-class= property **209**
    - in object definition 61, 65, 67-68
  - index-key= property **209**
    - in object definition 61, 65, 67-68
  - <index-key-reference> object **113**
    - in object summaries 143, 154
    - in parent content model 143
    - in property definition 222, 225
    - referenced 72, 113, 151
  - <index-page-citation-list> object **113**
    - in object summaries 143, 154
    - in property definition 217
    - referenced 113, 151
  - <index-page-citation-list-separator> object **113**
    - in object summaries 143, 154
    - in parent content model 143
    - referenced 113, 151
  - <index-page-citation-range-separator> object **113**
    - in object summaries 143, 154
    - in parent content model 143
    - referenced 113, 151
  - <index-page-number-prefix> object **113**
    - in object summaries 143, 154
    - in parent content model 143
    - referenced 113, 151

- `<index-page-number-suffix>` object **113**
  - in object summaries 143, 154
  - in parent content model 143
  - referenced 113, 151
- `<index-range-begin>` object **113**
  - in object summaries 143, 154
  - in property definition 209-210
  - referenced 113, 151
- `<index-range-end>` object **113**
  - in object summaries 143, 154
  - in property definition 225
  - referenced 113, 151
- `indicate-destination=` property **210**
- inherited properties 57
  - summary list 176-178
- `inherited-property-value()` function **138**
  - in function summaries 136
  - referenced 56
- `initial-page-number=` property **210**
  - in object definition 65
- `<initial-property-set>` object **81**
  - in object summaries 143, 154
  - in property definition 183-196, 199, 201-202, 205-207, 213-214, 219-221, 223-224, 226-233, 235-238
  - referenced 81, 151, 159-161, 163, 167
- `<inline>` object **81**
  - in object summaries 143, 154
  - in parent content model 143
  - in property definition 182-196, 199, 201-202, 205-212, 214, 219-221, 223-224, 226-228, 230-233, 235, 237-238
  - referenced 72, 81, 151, 159-161, 163, 166-167
- `<inline-container>` object **98**
  - in object summaries 144, 154
  - in property definition 182, 184-196, 199, 202, 208-212, 214, 218-221, 225-226, 230-231, 238-239
  - referenced 72, 98, 151, 161, 166-167
- `inline-progression-dimension=` property **210**
  - referenced 216
- `inline-progression-direction` 115
- `<instream-foreign-object>` object **87**
  - in object summaries 144, 154
  - in property definition 182-196, 199-202, 208-212, 214, 218-221, 223-224, 226-234, 237-238
  - referenced 72, 87, 151, 159-161, 166-167
- `<integer>` data type **169**
  - in property definition 169, 171, 187-188, 190, 194, 218, 238-239
  - referenced 225
- interactive objects 121-122
- `internal-destination=` property **210**
  - referenced 203, 210
- `intrinsic-scale-value=` property **210**
  - in inherited property summary 176
- `intrusion-displace=` property **211**
  - in inherited property summary 176
- J**
- K**
- `<keep>` data type **169**
  - in property definition 211-212
- `keep-together=` property **211**
  - in inherited property summary 176
  - referenced 222
- `keep-with-next=` property **211**
  - referenced 221
- `keep-with-previous=` property **211**
  - referenced 221
- Kleene operator 27
- L**
- `label-end()` function **138**
  - in function summaries 136
- landscape 106-107
- `<language>` data type **169**
  - in property definition 212
- `language=` property **212**
  - in inherited property summary 176
  - in object definition 65
  - in set of common properties 164
  - referenced 239
- `last-line-end-indent=` property **212**
  - in inherited property summary 177
- layout
  - layout-based formatting 45, 47
  - semantics 48

- `<layout-master-set>` object **63**
    - in object summaries 144, 153
    - in parent content model 61, 145
    - referenced ?, 46, 60, 149
  - `<leader>` object **87**
    - in object summaries 144, 154
    - in property definition 182-196, 199, 201-202, 205-207, 209-214, 219-221, 223-224, 226-228, 230-235, 237-238
    - referenced 72, 87, 151, 159-161, 163, 166-167, 212-213
  - `leader-align=` property **212**
    - in inherited property summary 177
  - `leader-length=` property **212**
    - in inherited property summary 177
  - `leader-pattern=` property **212**
    - in inherited property summary 177
    - referenced 228
  - `leader-pattern-width=` property **213**
    - in inherited property summary 177
  - leaders 86
  - `left=` property **213**
    - in set of common properties 158
    - referenced 228
  - `<length>` data type **170**
    - in property definition 170-171, 181, 184-186, 194, 196, 198-201, 203, 205, 208, 210, 212-213, 216-217, 222, 225, 227-229, 232, 234-236, 238
    - referenced 214-215, 218-221, 236
  - `<length-bp-ip-direction>` data type **169**
    - in property definition 193
  - `<length-conditional>` data type **170**
    - in property definition 187, 189, 191, 194, 218-220
  - `<length-range>` data type **170**
    - in property definition 186, 210, 212
  - `letter-spacing=` property **213**
    - in inherited property summary 177
  - `letter-value=` property **213**
    - in object definition 65
  - line
    - creating 115
    - elastic 83
  - `line-height=` property **213**
    - in inherited property summary 177
    - referenced 58, 210
  - `line-height-shift-adjustment=` property **214**
    - in inherited property summary 177
  - `line-stacking-strategy=` property **214**
    - in inherited property summary 177
  - `linefeed-treatment=` property **214**
    - in inherited property summary 177
    - referenced 237
  - link, see hyperlink
  - list formatting 128
  - `<list-block>` object **87**
    - in object summaries 144, 154
    - in property definition 183-198, 201-203, 209-212, 215, 219-221, 223-228, 230-233, 237
    - referenced 71, 87, 150, 159-161, 165, 167
  - `<list-item>` object **87**
    - in object summaries 144, 154
    - in parent content model 144
    - in property definition 183-197, 201-203, 209-212, 215, 219-221, 223-224, 226-228, 230-233, 237
    - referenced 87, 150, 159-161, 165, 167
  - `<list-item-body>` object **87**
    - in object summaries 144, 154
    - in parent content model 144
    - in property definition 209-211, 228, 230
    - referenced 87, 150, 159
  - `<list-item-label>` object **87**
    - in object summaries 144, 154
    - in parent content model 144
    - in property definition 209-211, 228, 230
    - referenced 87, 150, 159
  - lists of effective pages 47
  - loose-leaf publishing 47
- ## M
- `margin=` property **214**
    - in object definition 165
    - in property definition 215
    - in shorthand definition 179
  - `margin-bottom=` property **214**
    - in set of common properties 165
    - referenced 214

margin-left= property **215**  
     in set of common properties 165  
     referenced 214

margin-right= property **215**  
     in set of common properties 165  
     referenced 214

margin-top= property **215**  
     in set of common properties 165  
     referenced 214

<marker> object **104**  
     in object summaries 144, 155  
     in property definition 215  
     referenced 68, 104, 151, 215, 227

marker-class-name= property **215**  
     referenced 227

markup  
     syntax preservation 24

master-name= property **215**

master-reference= property **216**  
     in object definition 65  
     referenced 65

Mathematical Markup Language (MathML)  
     54

max() function **138**  
     in function summaries 136

max-height= property **216**

max-width= property **216**

maximum-repeats= property **216**

media-usage= property **216**  
     in object definition 61  
     referenced 61

merge-pages-across-index-key-references=  
     property **216**  
     in inherited property summary 177

merge-property-values() function **138**  
     in function summaries 136  
     referenced 56-57

merge-ranges-across-index-key-references=  
     property **217**  
     in inherited property summary 177

merge-sequential-page-numbers= proper-  
     ty **217**  
     in inherited property summary 177

min() function **138**  
     in function summaries 136

min-height= property **217**

min-width= property **217**

<multi-case> object **122**  
     in object summaries 144, 153  
     in parent content model 144  
     in property definition 197, 209-210, 228,  
     230, 232  
     referenced 122, 152, 159, 183, 197, 232-233

<multi-properties> object **122**  
     in object summaries 144, 153  
     in property definition 228, 230  
     referenced 73, 122, 152, 159, 181

<multi-property-set> object **122**  
     in object summaries 144, 153  
     in parent content model 144  
     in property definition 181, 209-210  
     referenced 122, 152

<multi-switch> object **122**  
     in object summaries 144, 153  
     in property definition 183, 209-210, 228,  
     230  
     referenced 73, 122, 152, 159, 183

<multi-toggle> object **122**  
     in object summaries 144, 153  
     in property definition 209-210, 228, 230,  
     233  
     referenced 72, 122, 152, 159, 197, 233

## N

<name> data type **170**  
     in property definition 197, 199, 204,  
     215-216, 225, 227, 233

namespaces 19, 31, 53-54  
     default namespace 31

navigation tools (as within a book) 12, 101

normal-flow-reference-area 115

<number> data type **170**  
     in property definition 169, 171, 199, 205,  
     208-210, 213, 216-218, 223, 227, 232-233,  
     237  
     referenced 232

number-columns-repeated= property **217**

number-columns-spanned= property **217**  
     referenced 199

number-rows-spanned= property **217**

## O

odd-or-even= property **218**

optimum 55  
 orphans= property **218**  
   in inherited property summary 177  
 overflow= property **218**  
**P**  
 padding= property **218**  
   in object definition 162  
   in property definition 219-221  
   in shorthand definition 179  
 padding-after= property **218**  
   in set of common properties 161  
 padding-before= property **219**  
   in set of common properties 161  
 padding-bottom= property **219**  
   in set of common properties 161  
   referenced 218  
 padding-end= property **219**  
   in set of common properties 161  
 padding-left= property **220**  
   in set of common properties 161  
   referenced 218  
 padding-rectangle 77  
 padding-right= property **220**  
   in set of common properties 161  
   referenced 218  
 padding-start= property **220**  
   in set of common properties 161  
 padding-top= property **221**  
   in set of common properties 161  
   referenced 218  
 page  
   choreography 103  
   dimensions 7  
   fidelity, see rendering; page fidelity  
   geometry 63, 79-80, 102, 106-110, 124  
   parity 102  
   position 102  
   region, see regions  
 page number citations 101  
 Page Sequence Master Interleave (PSMI)  
   106-110  
 page-break-after= property **221**  
   in property definition 197, 211  
   in shorthand definition 179

page-break-before= property **221**  
   in property definition 197, 211  
   in shorthand definition 179  
 page-break-inside= property **221**  
   in inherited property summary 177  
   in property definition 211  
   in shorthand definition 179  
 page-citation-strategy= property **222**  
 page-height= property **222**  
   referenced 216, 222, 229  
 <page-number> object **104**  
   in object summaries 144, 154  
   in property definition 182-196, 201-202,  
   205-207, 209-214, 219-221, 223-224,  
   226-238  
   referenced 72, 104, 149, 159-161, 163,  
   166-167  
 <page-number-citation> object **81**  
   in object summaries 144, 154  
   in property definition 182-196, 201-202,  
   205-207, 209-214, 219-221, 223-238  
   referenced 13, 72, 81, 151, 159-161, 163,  
   166-167  
 <page-number-citation-last> object **81**  
   in object summaries 144, 154  
   in property definition 182-196, 201-202,  
   205-207, 209-214, 219-238  
   referenced 72, 81, 151, 159-161, 163,  
   166-167  
 page-number-treatment= property **222**  
   in inherited property summary 177  
 page-position= property **222**  
 <page-sequence> object **65**  
   in object summaries 145, 153  
   in parent content model 61, 67, 145  
   in property definition 200, 204, 207-210,  
   212-213, 216, 225, 239  
   referenced ?, 46, 60, 68, 74-75, 149, 204,  
   216  
 <page-sequence-master> object **105**  
   in object summaries 145, 153  
   in parent content model 63, 144  
   in property definition 216  
   referenced 105, 149

`<page-sequence-wrapper>` object **67**  
 in object summaries 145, 153  
 in parent content model 61, 67, 145  
 in property definition 209-210  
 referenced 46, 60, 65, 67, 149  
`page-width=` property **222**  
 referenced 216, 222, 229  
 pagination 12, 100-110  
 semantics 13, 21, 26, 48, 51  
`pause=` property **222**  
 in object definition 160  
 in property definition 223  
 in shorthand definition 179  
`pause-after=` property **222**  
 in set of common properties 160  
 referenced 222  
`pause-before=` property **223**  
 in set of common properties 160  
 referenced 222  
`<percentage>` data type **170**  
 in property definition 170, 181-182,  
 184-186, 196, 199-200, 203, 205, 208,  
 210-213, 216-217, 222-223, 225, 227, 230,  
 232, 234-238  
 referenced 214-215, 218-221, 236  
 perimeter regions 79-80, 97  
`pitch=` property **223**  
 in inherited property summary 177  
 in set of common properties 160  
`pitch-range=` property **223**  
 in inherited property summary 177  
 in set of common properties 160  
`play-during=` property **224**  
 in set of common properties 160  
 portability problems 54, 58  
 Portable Document Format (PDF) 13, 32-34,  
 40, 49  
`position=` property **224**  
 in object definition 158, 167  
 in property definition 181, 226  
 in shorthand definition 179  
 precedence  
 areas 116  
`precedence=` property **224**  
 referenced 79  
 processing model 45, 50-52

properties 54  
 compound 58  
 direct vs. constraint 55  
 inherited 57, 176-178  
 property classes 45  
 shorthand 51, 58, 179  
 summary list 180-239  
 value expressions 56  
`proportional-column-width()` function  
**139**  
 in function summaries 136  
`provisional-distance-between-starts=`  
 property **224**  
 in inherited property summary 177  
`provisional-label-separation=` property  
**225**  
 in inherited property summary 177

## Q

## R

recommendations 35  
`ref-id=` property **225**  
`ref-index-key=` property **225**  
 reference orientation 78  
`reference-orientation=` property **225**  
 in object definition 65  
 referenced 138, 216  
 Refined Formatting Object Tree 51  
 refinement 51, 54  
`<region-after>` object **104**  
 in object summaries 145, 153  
 in parent content model 145  
 in property definition 184-196, 199,  
 202-203, 218-221, 224-225, 239  
 referenced 79, 104, 149, 161  
`<region-before>` object **104**  
 in object summaries 145, 153  
 in parent content model 145  
 in property definition 184-196, 199,  
 202-203, 218-221, 224-225, 239  
 referenced 79, 104, 149, 161



- `<region-body>` object **81**
  - in object summaries 145, 153
  - in parent content model 145
  - in property definition 184-196, 199, 202-203, 215, 218-221, 225, 230, 232, 239
  - referenced 79, 81, 149, 161, 165
- `<region-end>` object **104**
  - in object summaries 145, 153
  - in parent content model 145
  - in property definition 184-196, 199, 202-203, 218-221, 225, 239
  - referenced 79, 104, 149, 161
- `region-name=` property **225**
- `region-name-reference=` property **225**
- `<region-name-specifier>` object **105**
  - in object summaries 145, 153
  - in parent content model 143
  - in property definition 226
  - referenced 105, 149
- `<region-start>` object **104**
  - in object summaries 145, 153
  - in parent content model 145
  - in property definition 184-196, 199, 202-203, 218-221, 225, 239
  - referenced 79, 104, 149, 161
- regions 79-80, 97, 101
- `relative-align=` property **226**
  - in inherited property summary 177
  - referenced 202
- `relative-position=` property **226**
  - in set of common properties 167
  - referenced 224
- rendering 45, 48-49
  - agent 52
  - page fidelity 40
  - semantics 49
- `rendering-intent=` property **226**
- RenderX XEP 40, 54
- `<repeatable-page-master-alternatives>` object **105**
  - in object summaries 145, 153
  - in parent content model 145
  - in property definition 216
  - referenced 105, 149
- `<repeatable-page-master-reference>` object **105**
  - in object summaries 145, 153
  - in parent content model 145
  - in property definition 216
  - referenced 105, 149
- result tree 29
- `retrieve-boundary=` property **226**
- `retrieve-boundary-within-table=` property **226**
- `retrieve-class-name=` property **227**
- `<retrieve-marker>` object **104**
  - in object summaries 145, 155
  - in property definition 226-227
  - referenced 73, 104, 149, 227
- `retrieve-position=` property **227**
- `retrieve-position-within-table=` property **227**
- `<retrieve-table-marker>` object **104**
  - in object summaries 145, 155
  - in property definition 227
  - referenced 73, 104, 150, 227
- `rgb()` function **139**
  - in function summaries 136
- `rgb-icc()` function **139**
  - in function summaries 136
- `richness=` property **227**
  - in inherited property summary 177
  - in set of common properties 160
- `right=` property **227**
  - in set of common properties 158
- `role=` property **228**
  - in set of common properties 159
- roman numeral numbering 175
- `<root>` object **61**
  - in object summaries 145, 153
  - in property definition 209-210, 216, 228, 230
  - referenced ?, 46, 60, 63, 65, 67, 149, 159, 230
- `round()` function **139**
  - in function summaries 136
  - referenced 56
- rows (table)
  - spanning 92

rule-style= property **228**  
 in inherited property summary 177  
 rule-thickness= property **228**  
 in inherited property summary 177  
**S**  
 Scalable Vector Graphics (SVG) 19, 54, 83  
 scale-option= property **228**  
 in inherited property summary 177  
 scaling= property **228**  
 scaling-method= property **228**  
 <scaling-value-citation> object **87**  
 in object summaries 145, 154  
 in property definition 182-196, 200-202,  
 205-214, 219-221, 223-238  
 referenced 72, 87, 152, 159-161, 163,  
 166-167  
 score-spaces= property **229**  
 in inherited property summary 177  
 <script> data type **170**  
 in property definition 229  
 script= property **229**  
 in inherited property summary 177  
 in set of common properties 164  
 semantic information processing 18  
 sequence  
 sequence constructor, see template  
 serialization of result tree 29  
 server delivery 32-34  
 <shape> data type **170**  
 in property definition 198  
 shorthand properties 58  
 summary list 179  
 show-destination= property **229**  
 <simple-page-master> object **81**  
 in object summaries 145, 153  
 in parent content model 63, 144  
 in property definition 203, 215-216, 222,  
 225, 230, 232, 239  
 referenced 81, 124, 149, 165  
 <single-page-master-reference> object  
**105**  
 in object summaries 145, 153  
 in parent content model 145  
 in property definition 216  
 referenced 105, 149

size= property **229**  
 in property definition 222  
 in shorthand definition 179  
 source-document= property **229**  
 in set of common properties 159  
 <space> data type **170**  
 in property definition 213, 230, 238  
 space-after= property **230**  
 in set of common properties 165  
 space-before= property **230**  
 in set of common properties 165  
 referenced 53, 55, 198  
 space-end= property **230**  
 in set of common properties 166  
 space-start= property **230**  
 in set of common properties 166  
 span= property **231**  
 spanning  
 table columns/cells, see columns (table);  
 spanned  
 table rows, see rows (table); spanning  
 speak= property **231**  
 in inherited property summary 177  
 in set of common properties 160  
 speak-header= property **231**  
 in inherited property summary 177  
 in set of common properties 160  
 speak-numeral= property **231**  
 in inherited property summary 177  
 in set of common properties 160  
 speak-punctuation= property **231**  
 in inherited property summary 177  
 in set of common properties 160  
 speech-rate= property **232**  
 in inherited property summary 177  
 in set of common properties 160  
 src= property **232**  
 stacking 70, 97, 115  
 Standard Page Description Language (SPDL)  
 49  
 start-indent= property **232**  
 in inherited property summary 177  
 in set of common properties 165  
 referenced 57  
 starting-state= property **232**  
 starts-row= property **232**

static content 80, 101

`<static-content>` object **104**  
     in object summaries 145, 153  
     in parent content model 65, 145  
     in property definition 204, 209-210  
     referenced 104, 149

`stress=` property **233**  
     in inherited property summary 177  
     in set of common properties 160

`<string>` data type **171**  
     in property definition 197, 200, 207, 209,  
     225, 228, 234  
     referenced 234

styling structured information 23, 27

`suppress-at-line-break=` property **233**

`switch-to=` property **233**

`system-color()` function **139**  
     in function summaries 136

`system-font()` function **139**  
     in function summaries 136  
     referenced 204

## T

`<table>` object **93**  
     in object summaries 146, 154  
     in parent content model 146  
     in property definition 183-198, 201-203,  
     208-212, 215, 219-221, 223-224, 226-228,  
     230-233, 237-239  
     referenced 71, 93, 150, 159-161, 165, 167

`<table-and-caption>` object **93**  
     in object summaries 146, 154  
     in property definition 183-198, 201-203,  
     209-212, 215, 219-221, 223-224, 226-228,  
     230-234, 237  
     referenced 71, 93, 150, 159-161, 165, 167

`<table-body>` object **93**  
     in object summaries 146, 154  
     in parent content model 146  
     in property definition 183-196, 201-202,  
     209-210, 219-221, 223-224, 226-228,  
     230-233, 237  
     referenced 93, 150, 159-161, 167

`<table-caption>` object **93**  
     in object summaries 146, 154  
     in parent content model 146  
     in property definition 183-196, 201-202,  
     208-211, 219-221, 223-224, 226-228,  
     230-233, 237-238  
     referenced 93, 150, 159-161, 167

`<table-cell>` object **93**  
     in object summaries 146, 154  
     in parent content model 146  
     in property definition 183-196, 199,  
     201-203, 208-210, 217-221, 223-224,  
     226-228, 230-233, 237-238  
     referenced 93, 150, 159-161, 167, 199, 217,  
     234

`<table-column>` object **93**  
     in object summaries 146, 154  
     in parent content model 146  
     in property definition 184-196, 199, 217,  
     219-221, 237  
     referenced 93, 138, 150, 161, 199, 217

`<table-footer>` object **93**  
     in object summaries 146, 154  
     in parent content model 146  
     in property definition 183-196, 201-202,  
     209-210, 219-221, 223-224, 226-228,  
     230-233, 237  
     referenced 93, 150, 159-161, 167, 233

`<table-header>` object **93**  
     in object summaries 146, 154  
     in parent content model 146  
     in property definition 183-196, 201-202,  
     209-210, 219-221, 223-224, 226-228,  
     230-233, 237  
     referenced 93, 150, 159-161, 167, 233

`table-layout=` property **233**

`table-omit-footer-at-break=` property  
**233**

`table-omit-header-at-break=` property  
**233**

- <table-row> object **93**
  - in object summaries 146, 154
  - in parent content model 146
  - in property definition 183-197, 201-202, 208-212, 219-221, 223-224, 226-228, 230-233, 237
  - referenced 93, 150, 159-161, 167, 197
- tables 89-93
- tables of content 83
- target-presentation-context= property **233**
- target-processing-context= property **234**
- target-stylesheet= property **234**
- template 24-25
- TeX 49
- text-align= property **234**
  - in inherited property summary 177
- text-align-last= property **234**
  - in inherited property summary 177
- text-altitude= property **234**
- text-decoration= property **234**
  - referenced 57
- text-depth= property **235**
- text-indent= property **235**
  - in inherited property summary 177
- text-shadow= property **235**
- text-transform= property **235**
  - in inherited property summary 177
- <time> data type **171**
  - in property definition 222-223
- <title> object **81**
  - in object summaries 146, 153
  - in parent content model 65, 145
  - in property definition 183-196, 199, 201-202, 205-207, 214, 219-221, 223-224, 227-228, 230-233, 237
  - referenced 81, 149, 159-161, 163, 166
- top= property **236**
  - in set of common properties 158
- traits 51, 53
  - formatting 51
  - rendering 51
- transforming information 23
- treat-as-word-space= property **236**
- tuples 89
- Turing complete 24
- typographical conventions 2
- U**
- Unicode 175
- unicode-bidi= property **236**
- Universal Character Set (UCS), see Unicode
- Universal Resource Identifier 19
- <uri-specification> data type **171**
  - in property definition 184, 200-201, 203, 224, 228-229, 232, 234
  - referenced 224, 228
- V**
- vendor questions 241-243
- vertical-align= property **236**
  - in property definition 182, 186, 202
  - in shorthand definition 179
- visibility= property **236**
  - in inherited property summary 177
- visual media 26, 59
- vocabulary, XML 10, 18-19
- voice-family= property **237**
  - in inherited property summary 177
  - in set of common properties 160
- volume= property **237**
  - in inherited property summary 177
  - in set of common properties 160
- W**
- W3C XSL Working Group 23
- white-space= property **237**
  - in inherited property summary 177
  - in shorthand definition 179
- white-space-collapse= property **237**
  - in inherited property summary 177
  - referenced 237
- white-space-treatment= property **238**
  - in inherited property summary 177
  - referenced 237
- widows= property **238**
  - in inherited property summary 177
- width= property **238**
  - referenced 213
- word-spacing= property **238**
  - in inherited property summary 177
- wrap-option= property **238**
  - in inherited property summary 178
  - referenced 237

**<wrapper> object 81**

- in object summaries 146, 155
- in parent content model 144
- in property definition 209-210
- referenced 73, 81, 150, 152

writing direction 78

writing mode, see writing direction

**writing-mode= property 238**

- in inherited property summary 178
- in object definition 65
- referenced 138, 201, 216

WSSSL 23

**X**

XML, see Extensible Markup Language (XML)

XML processor 24

**xml.lang= property 239**

- in inherited property summary 178
- in object definition 65, 164
- in property definition 200, 212
- in shorthand definition 179

xml:space 18

XSL, see Extensible Stylesheet Language (XSL)

XSL-FO, see Extensible Stylesheet Language Formatting Objects (XSL-FO)

XSL-FO processor 26

**<xsl:attribute> (XSLT) 127**

XSLT, see Extensible Stylesheet Language Transformations (XSLT)

XSLT processor 24

**Y**

**Z**

**z-index= property 239**

ZIP file of examples 5